FORESTRY AND NATURAL SCIENCES

ANU RANINEN

Practical Process Improvement

How to Initiate Software Process Improvement in a Small Company

PUBLICATIONS OF THE UNIVERSITY OF EASTERN FINLAND
Dissertations in Forestry and Natural Sciences

UNIVERSITY OF EASTERN FINLAND

ANU RANINEN

Practical Process Improvement

How to Initiate Software Process Improvement in a Small Company

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences
No 136

Academic Dissertation
To be presented by permission of the Faculty of Science and Forestry for public examination in the Auditorium ML2 in Medistudia Building at the University of Eastern Finland, Kuopio, on April, 25, 2014,
at 12 o'clock noon.

School of Computing

Kopijyvä
Kuopio, 2014
Editors: Profs. Pertti Pasanen, Pekka Kilpeläinen, Kai Peiponen,
and Matti Vornanen

Distribution:
University of Eastern Finland Library / Sales of publications
P.O. Box 107, FI-80101 Joensuu, Finland
tel. +358-50-3058396
http://www.uef.fi/kirjasto

ISBN: : 978-952-61-1429-3 (printed)
ISSNL: 1798-5668
ISSN: 1798-5668
ISBN: 978-952-61-1430-9 (pdf)
ISSNL: 1798-5668
ISSN: 1798-5668

Author's address: University of Eastern Finland
School of Computing
P.O.Box 1627
70211 KUOPIO
FINLAND
email: anu.m.raninen@gmail.com

Supervisors: Professor Jarmo J. Ahonen, Ph.D.
University of Eastern Finland
School of Computing
P.O.Box 1627
70211 KUOPIO
FINLAND
email: jarmo.ahonen@uef.fi

Research fellow Sarah Beecham, Ph.D.
University of Limerick
Lero – The Irish Software Engineering Centre
University of Limerick
Limerick
IRELAND
email: sarah.beecham@lero.ie

Reviewers: Professor Markku Oivo, Ph.D.
University of Oulu
Department of Computer Science
P.O.Box 3000
90014 Oulu
FINLAND
email: markku.oivo@oulu.fi

Principal lecturer Nathan Baddoo, Ph.D.
University of Hertfordshire
School of Computer Science
Hatfield
Hertfordshire
AL10 9AB
UNITED KINGDOM
email: n.baddoo@herts.ac.uk

Opponent: Reader in Software Engineering Tracy Hall, Ph.D.
Brunel University
St John's 049
Uxbridge
UB8 3PH
UNITED KINGDOM
email: tracy.hall@brunel.ac.uk

ABSTRACT

Nowadays, when software has become part of most everyday appliances, the significance of the software industry and especially small software companies is growing. The vast majority of companies world-wide are Small or Medium-sized Enterprises (SMEs). In Finland, over half of the software companies have only two to ten employees.

Improving the operations of a company is necessary in order to maintain its competitive edge. Software Process Improvement (SPI) activities have been reported to result in remarkable improvements in the quality of software, reduced time to market and increased productivity.

However, despite the evidence suggesting that the size of the organization does not limit its potential for SPI success, small companies (SCs) have been somewhat neglected by the SPI research community. The techniques and standards developed to support SPI are generally aimed at the larger organizations. As a result, small software companies struggle to initiate their improvement projects.

The aim of this thesis is to provide a validated, practical, and easy to apply approach for small software companies to cost-effectively initiate SPI. The research is driven by close collaboration with SMEs undergoing SPI. Several case studies were conducted to develop and validate such an approach.

The main contribution of the thesis is LAPPI, a light-weight technique to practical process modeling and improvement target identification. The LAPPI technique enables small software companies to initiate SPI cost-effectively. LAPPI is complemented by research into motivation monitoring, a recommended practice that supports the initiation and success of SPI initiatives in SCs.

The contribution of the thesis is beneficial for stakeholders of SPI projects in small companies. The results are especially applicable in cases where companies are having trouble initiating SPI or have concerns about the cost of implementing SPI.

Universal Decimal Classification: 004.41, 004.413, 005.62, 334.713 INSPEC Thesaurus: software engineering; software process improvement; software quality; small-to-medium enterprises; human factors

Yleinen suomalainen asiasanasto: ohjelmistoala; ohjelmistokehitys; ohjelmistotuotanto; tietotekniikkayritykset; pienyritykset; prosessit; kehittäminen; mallintaminen; motivaatio

ACKNOWLEDGMENTS

This work has been carried out in University of Eastern Finland's School of computing over a period of 2006-2014. The research has been funded by TEKES - the Finnish Funding Agency for Technology and Innovation, and participating software companies in projects PraSEMO, SaMET and METRI. In addition, Lero - The Irish Software Engineering Centre and SFI - Science Foundation Ireland have supported the research during 2011, while I worked at Lero as a visiting researcher. I wish to thank all these organizations for their financial support.

Further, I wish to thank my supervisors: Professor Jarmo J. Ahonen for his continuous support and apt criticism, and Dr. Sarah Beecham, my supervisor from Lero, who has taught me so much about how to be a good researcher (and how to have fun at it).

Dr. Nathan Baddoo and Professor Markku Oivo I would like to thank for accepting the role of reviewer of my thesis and their valuable feedback. To Dr. Tracy Hall I want to express my utmost gratitude for her role as the opponent at my thesis public examination.

I also want to thank my co-authors; Hanna-Miina, Paula, Helena, Tanja and Lauri. In addition, I wish to thank every single one of my colleagues in Finland and in Ireland. It has been an absolute pleasure working with you. Especially, I wish to thank my office mates at Lero in 2011. Further, Dr. John Noll from Lero deserves my thanks. My thesis is so much better thanks to you guys.

In addition to my academic associates, I owe thanks to the employees of companies collaborating with the research presented in the thesis. It has been my honor working with you these past years. Seldom does a researcher get such instant and expert feedback on her work.

Last but definitely not least, I want to thank my family. My warmest thanks go to my parents Esko and Irmeli. Dad, here it is, finally, for you to read. Mum, you're the best; I can always count on you. Hope I've made you proud. Further, my little sister Sini deserves my gratitude. There is a lot to learn from her persistent attitude towards getting what you want, in the academic world as well as in life. My grandma, Terttu, I want to thank for her positive attitude and support. Further, I shouldn't forget thanking Reino and Iivari, the most well-loved members of our pack. My final thanks goes to my husband, Marko, who has had the debatable pleasure of following the writing process of his whirlwind wife's thesis up-close. Thanks for your love and patience.

Kuopio March 19, 2014

Anu Raninen

LIST OF PUBLICATIONS

This thesis consists of the present review of the author's work in the field of process improvement in small software companies and the following selection of the author's publications:

I A. Valtanen and J.J. Ahonen, "Big Improvements with Small Changes: Improving the Processes of a Small Software Company," *Product-Focused Software Process Improvement, Lecture Notes in Computer Science* **5089,** 258–272 (2008).

II A. Valtanen and H-M. Sihvonen, "Employees' Motivation for SPI: Case Study in a Small Finnish Software Company," *Software Process Improvement, Communications in Computer and Information Science* **16,** 152-163 (2008).

III A. Raninen, J.J. Ahonen, H-M. Sihvonen and P. Savolainen, "LAPPI: A Light-Weight Technique to Practical Process Modeling and Improvement Target Identification," *Journal of Software: Evolution and Process*, **25,** Issue 9, 915-933, (2013)

IV A. Raninen, H. Merikoski, J.J. Ahonen and S. Beecham, "Applying Software Process Modeling to Improve Customer Support Processes," Submitted to *Journal of Software: Evolution and Process*.

V T. Toroi, A. Raninen and L. Väätäinen, "Identifying Process Improvement Targets in Test Processes: A Case Study," *IEEE International Conference on Software Maintenance 2013*, DOI 10.1109/ICSM.2013.12, (2013)

Throughout the overview, these papers will be referred to by Roman numerals. The original publications are reprinted at the end of this thesis with permission of the copyright holders.

ABBREVIATIONS

CMMI©	Capability Maturity Model Integration
ICT	Information and Communications Technology
IEC	the International Electrotechnical Commission
ISO	the International Organization for Standardization
LAPPI	A light-weight technique to practical process modeling and improvement target identification
SC	Small Company, company with no more than 50 employees [1]
SE	Software Engineering
SEI	Software Engineering Institute
SME	Small or medium sized enterprise, a company with no more than 250 employees [1].
SPI	Software Process Improvement
SPICE	Software Process Improvement and Capability dEtermination
SW-CMM©	SoftWare Capability Maturity Model
VSE	Very Small Entity ”can mean an independent partnership or linked organization having up to 25 people that is engaged in a software implementation project.” [2]

GLOSSARY

Case organization(s) The companies where the research presented in the studies of the thesis were conducted. Studies I, II and IV have one case organization, whereas in studies III and V there are several case organizations.

Motivation monitoring The motivational interviews and the survey presented in study II [3].

Process ”A set of actions, tasks, and procedures that when performed or executed obtain a specific goal or objectives” [4].

Process overview The result of applying the LAPPI process modeling technique [5]; a document that combines steps, roles, information flows, problems and improvement suggestions related to a modeled process.

Software process ”The set of activities, methods, and practices used in the production and evolution of software” [6].

Software Process Improvement (SPI) A set of activities that will lead to a better software process, and through which higher quality software products will be delivered in a more timely manner.

Tacit knowledge ideas, discussions, awareness and accumulated learning experienced. Tacit knowledge has an important role to play in software SMEs SPI [7].

AUTHOR'S CONTRIBUTION

The publications selected in this dissertation are original research papers on software process improvement in small companies. Please note that the author has changed her surname in 2009 from Valtanen to Raninen.

I Anu Valtanen planned and conducted the research presented in collaboration with the case organization. Jarmo J. Ahonen participated in the planning process of the research and acted as the chairman of the process modeling sessions. The paper was written by Anu Valtanen and commented by Jarmo J. Ahonen.

II Anu Valtanen and Hanna-Miina Sihvonen have conducted the study and the writing process of the paper in collaboration.

III All of the authors have participated in the evolution process of the LAPPI technique. Anu Raninen has written most of the text in the paper. Jarmo J. Ahonen has written parts of Section 2 and helped by commenting on the other sections. Hanna-Miina Sihvonen has helped with writing Sections 1 and 2, and commented on the paper in general. Paula Savolainen has commented and enhanced the paper. Sarah Beecham supported the writing process and provided valuable comments on how to present the LAPPI technique.

IV The study was planned and implemented by Anu Raninen. The text in the article is also mainly written by her. Jarmo J. Ahonen acted as the chairman of the LAPPI process modeling sessions. Anu Raninen generated the customer satisfaction survey in collaboration with the target company and analyzed the first results from 2008. Helena Merikoski helped to analyze the customer satisfaction survey from 2009. Sarah Beecham conducted the statistical analysis presented, wrote section 4, and supported the overall writing process of the paper.

V Anu Raninen was responsible for designing the study presented in the paper. Further, she acted as the chairman of the LAPPI modeling sessions described in the paper and analyzed the results in collaboration with Tanja Toroi. Tanja Toroi documented the process modeling efforts in companies A and C. Lauri Väätäinen documented the modeling efforts implemented in company B. Anu Raninen wrote most of 1 Introduction, 5 Discussion and 6 Conclusion sections and commented on the other sections. Tanja Toroi wrote the remaining sections.

Contents

1 Introduction

This thesis aims to make initiating Software Process Improvement (SPI) projects easier in small software companies. Understanding the current state of the software processes and their problem points is crucial for the success of SPI initiatives [8,9]. Without this understanding, SPI resources may be allocated to less meaningful targets.

The vast majority of companies world-wide are small or medium sized enterprises (SMEs), with no more than 250 employees [10]. In Finland, the majority of software companies are even smaller, with no more than 50 employees [11]. Over half of these small companies (SCs) have merely two to ten employees [11].

Nowadays, when software has become part of most everyday appliances, the significance of the software industry and especially small software companies is large and growing [12, 13]. Among other things, small software companies ensure healthy competition. In order to maintain their competitiveness small companies need to continuously improve their operations [13]. Effective implementation of SPI approaches can help to achieve this goal [6].

The idea of better quality software products through better quality processes was developed in the 1980s [6]. Since then the SPI research community has been committed to process improvement research, see e.g. [14, 15]. In addition, there have been initiatives for standardization of SPI, e.g. ISO/IEC 15504 [16]. ISO/IEC 15504 among SEI's (Software Engineering Institute) CMMI© (Capability Maturity Model Integration) [17] are nowadays popular approaches for SPI. There are a lot of examples on the success of SPI initiatives, see for example [18–20].

However, despite the fact that it is shown that the size of the organization does not limit its potential for SPI success [21], small software companies have been neglected by the SPI research community for a long time. Only in recent years has the research focus shifted to include them [13]. Small companies can also achieve high organizational performance via SPI activities [21]. Further, in SMEs, a positive correlation between increases in SPI and increases in business success has been shown to exist [18]. However, the SPI programs intended for small companies should not only be scaled down versions of those applied in large companies [22].

Features that make small companies different from their large competitors, and in turn affect how their SPI programs should be run, have been researched extensively, see e.g. [7, 13, 23, 24]. Different kinds of process improvement approaches have been developed accordingly [25–28]. Perhaps the main challenge of SPI in small software companies is their often limited resources, time and money [13]. In addition, it may be difficult to maintain flexibility and adaptability in small software companies operations while improving process performance and maturity [21, 29].

Small software companies' SPI programs often rely on making small adjustments to the software process on a regular basis, rather than occasionally implementing larger SPI initiatives [7]. Further, process improvements in the smaller companies are often of a tacit nature: ideas, discussions, awareness and accumulated learning experienced [7]. In addition, cost-effectiveness and the ability to support process optimization are important qualities of an SPI approach suited to SMEs [7, 13, 22]. However, current process improvement standards usually address process optimization only on higher management levels [7]. High maturity is difficult to reach for small companies [23].

The majority of the process improvement approaches developed for small software companies use are based on process improvement standards like ISO/IEC 15504 [16] or maturity models like the CMMI© [17]. The approaches most often apply process assessment to visualize the current state of processes and initiate process improvement. Small companies often find it problematic to apply such approaches [2, 23]. The main problem often being that they consider the approaches too heavy weight for their purposes [2].

Further, smaller companies struggle with applying process assessments [7, 30] needed to visualize the current state of processes prior to applying the process improvement standards. The process assessment methods usually require more resources than small companies can afford to invest in order to produce meaningful results [30]. In addition, process assessment methods are not ideal for detecting tacit knowledge [7]. Further, small scale, informal process improvements are often difficult to measure and likely to be overlooked by conventional process assessments [7].

Hence, there is a lack of well focused, cost-effective, easy to adopt approach for initiating SPI in small companies. In addition, the approach should be suitable for process optimization and visualizing improvements in tacit knowledge. This thesis aims to fill this gap. The research problem of the thesis is: ”How to practically and cost-effectively initiate process improvement in a small software company?”

The main contribution of the thesis is the LAPPI (A light-weight technique to practical process modeling and improvement target identification) technique [5]. The LAPPI technique is a practical technique for process modeling developed in close collaboration with software practitioners working for small and medium-sized software companies. Applying LAPPI, software organizations can quickly model processes, make them visible, identify the problem points, and recognize undefined parts of the process. Further, LAPPI supports visualizing the tacit knowledge and points of improvement related to it [31].

In relation to the Initiating - Diagnosing - Establishing - Acting - Leveraging (IDEAL) model [32], which can be used as a roadmap for an SPI program, LAPPI is most beneficial in the diagnosing phase of SPI. The LAPPI technique consists of modeling workshops and documentation, is not tied to any SPI framework, and enables the use of different kinds of SPI approaches, for example, CMMI [17] and ISO/IEC 15504 [16]. LAPPI has been validated via over 40 case studies presented in [5,31,33].

Further, to complement the LAPPI technique, motivation monitoring is suggested to be applied to support the success and continuation of SPI initiatives [3]. Previous research has shown that management and employee motivation is needed to enable successful process improvement [34, 35]. Hence, motivation for SPI is important, especially in small software companies where resources are often limited. The small companies often cannot afford to employ dedicated process improvement personnel and existing employees, who are already burdened with their current duties, find it difficult to take on additional tasks such as SPI responsibilities [13].

The research presented in the thesis is based on five studies conducted in collaboration with the software industry. The studies present the LAPPI technique's different evolution phases (studies I and III), discuss the motivational aspects of SPI (study II) and validate the LAPPI technique (Studies IV and V). Each of the studies is published as an individual research paper. The papers can be seen as appendices of the thesis.

2 Research Background

The following subsections present a literature review of research related to the thesis. Section 2.1 describes the qualities of small companies. Section 2.2 describes the main characteristics of software process improvement. The most important standards and frameworks from the SPI perspective are presented in Section 2.3, highlighting the most relevant ones; the CMMI© [17], ISO/IEC 15504 [16], and ISO/IEC 29110 [2]. In section 2.4, motivational issues are covered from an SPI perspective. Section 2.5 concentrates on SPI in small companies. Section 2.6 presents process modeling and how it supports process improvement.

2.1 SMALL COMPANIES

"A small company" (SC) is defined as a company with less than 50 employees [1]. Very small entities (VSEs) are even smaller. "VSE can mean an independent partnership or linked organization having up to 25 people that is engaged in a software implementation project." [2]. Small and very small software companies are extremely important to the growth of many national economies [12, 13]. A significant amount of companies throughout Europe are small or very small. For example, in 2008 the majority of Finnish software companies employed less than 5 people [11]. The whole of Europe's small and medium sized companies (up to 250 employees) represent 93% of all business while in the US 56% of the businesses fall into this category [36].

Small companies have the same need to maintain and improve their competitiveness as their larger counterparts and consequently small companies also need to improve their processes. However, small software companies do not necessarily share the same characteristics and goals as large companies [13], which affects their SPI initiatives [23]. There are certain unique features of small companies that need to be understood. Their resources, both financial and human, are often limited, and management, work, and organizational culture may differ greatly from the ones in large organizations. Small companies are characterized by low hierarchy organization which enables direct communication. Hence, small companies are generally very flexible and reactive. In addition, small companies' employees often assume various roles in parallel [13, 24].

Further, small companies often apply agile software development methods [37]. According to Erickson et al. [38] ”agility means to strip away as much as of the heaviness, commonly associated with the traditional software development methodologies, as possible to promote quick response to changing environments, changes in user requirements, accelerated project deadlines and the like”. Agile methods can be seen as a reaction to traditional software development methods, which emphasize ”a rationalized, engineering-based approach” [37, 39, 40]. Agile methods are reported to best work on small teams, and thus are popular and often most beneficial in small companies [37].

2.2 SOFTWARE PROCESS IMPROVEMENT

Schedule and budget overruns are common in software industry [41, 42]. Furthermore, software products are often delivered with insufficient or unwanted functionality, poor reliability etc [43]. All these issues are related to the quality of the software. Watts Humphrey [6] developed the idea that the quality of a software system can be improved by improving the quality of the process used to develop it. Software process improvement (SPI) is based on these underlying principles.

Software process improvement is said to ”characterize all actions undertaken to improve an organization's processes to both increase their efficiency and meet the organization's business goals” [2]. Software process improvement is a set of activities that will lead to a better software process, and through which higher quality software products can be delivered in a more timely manner. Process improvement should be driven by business goals such as increasing productivity, customer satisfaction or increasing market share [2]. Plenty of evidence exists to show that improvement initiatives provide profitable results [19, 20, 44–46].

To initiate process improvement initiatives in a company, a process improvement environment is needed. Software process infrastructure, process improvement roadmap, process assessment method, and software process improvement plan are key components of this environment [9]. Software process infrastructure consists of the organizational and management infrastructure and the technical infrastructure. Improvement roadmap specifies steps towards realization of an effective software process, e.g. approaches like CMMI© and ISO/IEC 15504 can be used as such [9]. In addition, software process improvement program model, e.g. IDEAL (Initiating-Diagnosing-Establishing-Acting-Leveraging), can be used to guide development of a long-range, integrated plan for initiating and managing a SPI program [32].

Software process improvement is often approached through process assessment [6, 9, 30]. Process assessments help to make the strengths and weaknesses of the organization's processes visible and, thereby act as a catalyst for the SPI initiative and generation of the improvement plan [9, 30]. CMMI© and ISO/IEC 15504 have their own assessment models and in cases when those are not suitable a large variety of approaches can be found, see e.g. Adept [25] or METvalCOMPETISOFT [47]. The improvement plan is generated based on the process assessment results. In the improvement plan, the assessment findings are generated into specific improvement actions [9].

Several factors can affect the success of SPI programs. Successful process improvement starts at the top of the organization [9]. Senior management commitment is required to launch a change effort and to provide continuing resources. However, it is not enough to ensure the management's commitment, to create a successful and continuous improvement program the involvement of the whole organization is needed [9]. In addition to management and staff motivation, other important SPI success factors are clear and relevant SPI goals, staff time and adequate resources [35]. Furthermore, to make process improvement successful it should be understood as a continuous effort involving learning and evolution, not a one-off experiment [48]. In addition, the extent to which the software organization collects and utilizes quality data to guide and assess the effects of SPI activities has an impact on the SPI programs success [21]. Hence, a company should develop a measurement program in conjunction with their SPI initiatives [9].

2.3 PROCESS IMPROVEMENT STANDARDS AND FRAMEWORKS

There are numerous standards developed to ease the efforts in improving the software product quality. IEEE Standard Glossary of Software Engineering Terminology [49] defines standard as ”mandatory requirements employed and enforced to prescribe a disciplined uniform approach to software development, that is, mandatory conventions and practices are in fact standards”. Another definition given by the ISO/IEC Guide [50] is that a standard is ”a document, established by consensus and approved by a recognized body that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context”.

Customs, conventions, company products, corporate standards, etc. sometimes become generally accepted and dominant; these kinds of documentations are called ”de-facto standards” [51]. In addition to ”standards” and ”de-facto standards”, ”framework” is a much used term when covering sets of documentation defining different aspects of software engineering in a standard like manner. The Oxford English Dictionary [52] defines framework as ”a basic structure underlying a system, concept, or text”.

Figure 2.1 presents the standards and frameworks most relevant to this research and the relationships between them. The Figure

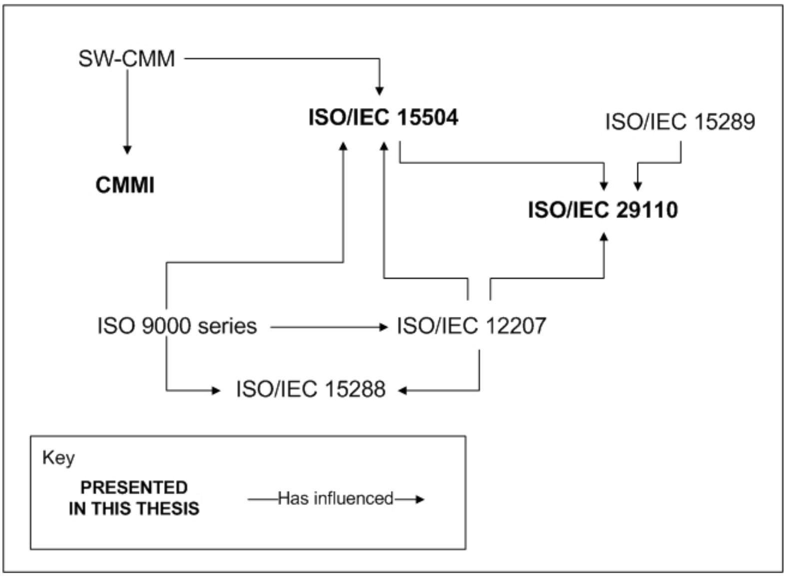

Figure 2.1: Standards and frameworks related to the research [53].

adapts that of Sheard [53].

SPI has its roots in quality management and improvement [54] fields. ISO 9000 [55] standard series covers quality management and quality improvement is discussed for example in Six Sigma [56]. Six Sigma is a set of techniques, and tools for process improvement developed. Six Sigma is not presented in figure 2.1 but has influenced the development of CMMI and hence, the development of ISO/IEC 15504 [57].

ISO 9001 has also had an influence in the evolution of ISO/IEC 15504 [16]. ISO/IEC 15504 aka "SPICE" (Software Process Improvement and Capability dEtermination) is a framework for the assessment of processes and was initially derived from ISO/IEC 12207 and from maturity models like Bootstrap [14] and the SW-CMM© [15]. ISO/IEC 15504 and CMMI© (Capability Maturity Model Integration) [17], the successor of SW-CMM©, seem to be the most popular SPI frameworks nowadays, see e.g. [58].

ISO/IEC 29110 [2] was conceived due to small companies having serious trouble adopting ISO/IEC 12207 [59] and ISO/IEC 15289

[60]. ISO/IEC 12207 is a software lifecycle processes standard that aims to define all the tasks required for developing and maintaining software. ISO/IEC 15289 was developed to assist users of systems and software life cycle processes to manage information items (usually meaning documentation) and is based on the life cycle processes described in ISO/IEC 12207 or ISO/IEC 15288 [61]. ISO/IEC 29110 is planned for very small entities (enterprises, organizations, departments or projects) (VSEs), i.e. companies employing no more than 25 people. The purpose of the ISO/IEC 29110 standard is to define a subset of ISO/IEC standards relevant to the VSE context, i.e. processes and outcomes of ISO/IEC 12207 and products of ISO/IEC 15289. ISO/IEC 29110 also has a link to ISO/IEC 15504. ISO/IEC 15504's process assessment requirements are applied also in ISO/IEC 29110.

CMMI© and ISO/IEC 15504 set the foundation of contemporary process improvement work. These two have not been directly used in this thesis but they have provided the reference model for the research presented. These two and ISO/IEC 29110 are described in more detail in the next subsections.

2.3.1 The CMMI©

The CMMI© (Capability Maturity Model Integration) [17] is a process improvement approach developed by the SEI (Software Engineering Institute) that provides organizations with the essential elements of effective processes that aim to improve their performance. The CMMI© can be used to guide process improvement across a project, a division, or an entire organization. It helps integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes. The CMMI© is being adopted worldwide. [62]

The CMMI© can be used in three different areas:

- Product and service development (CMMI© for Development model)

- Service establishment, management, and delivery (CMMI© for Services model)
- Product and service acquisition (CMMI© for Acquisition model)

CMMI© can be applied according to two representations; continuous and staged. The continuous representation allows the user to focus on the specific processes that are considered important for the organization's immediate business goals. The staged representation is designed to provide a standard sequence of improvements, and can serve as a basis for comparing the maturity of different projects and organizations. Applying the staged representation, one can achieve maturity levels from one to five; 1-Initial, 2-Managed, 3-Defined, 4-Quantitatively Managed, and 5-Optimizing. [62]

In CMMI©, the maturity of a process measured via process appraisals. The Standard CMMI Appraisal Method for Process Improvement (SCAMPI) is the official CMMI© assessment method used to evaluate organizations processes and provide ratings of maturity levels [63].

2.3.2 ISO/IEC 15504

ISO/IEC 15504 [16], also known as SPICE (Software Process Improvement and Capability dEtermination), is a process assessment framework. It was initially developed based on software life cycle processes standard ISO/IEC 12207 and from maturity models like Bootstrap and the SW-CMM©. ISO/IEC 15504 is an international standard.

ISO/IEC 15504 provides a structured approach for the assessment of processes. The assessments can be done inside a company to better understand its' own processes, to meet requirements of a customer or other stakeholder or for benchmarking purposes. The framework for process assessment "facilitates self-assessment, provides a basis for use in process improvement and capability determination, takes into account the context in which the assessed process is implemented, produces a process rating, addresses the ability of the process to achieve its purpose, is appropriate across all application domains and sizes of organization and may provide an objective benchmark between organizations". ISO/IEC 15504 is composed of five parts. The parts are illustrated in Figure 2.2.

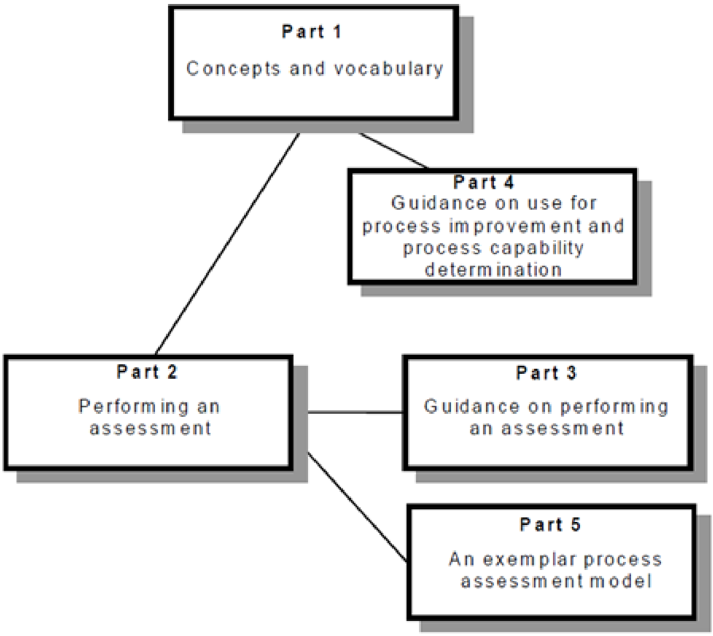

Figure 2.2: Components of ISO/IEC 15504 [16].

ISO/IEC 15504 defines six levels to assess organizations' process capability: 0 - Incomplete process, 1 - Performed process, 2 - Managed process, 3 - Established process, 4 - Predictable process, and 5 - Optimizing process [16]. ISO/IEC 15504 part 2 defines the requirements for performing process assessment as a basis for use in process improvement and capability determination.

The ISO/IEC 15504 standard series will be replaced and evolved further by 330xx standard series in the near future [64]. ISO/IEC TR 33014:2013 Process assessment – Guide for process improvement has already been published as a technical report [65]. ISO/IEC

33014:2013 replaces ISO/IEC 15504:4. However, the new standards were not publicly available while conducting the research presented.

2.3.3 ISO/IEC 29110

ISO/IEC 29110 is an emerging standard initiated in 2005. The rationale to develop such a standard is that the majority of ISO/IEC standards do not address the needs of VSEs [2, 23]. ISO/IEC 29110 standard's evolution began in May 2005, at the SC7 Plenary Meeting in Finland, where a resolution was approved to ballot a proposal for the development of software life-cycle profiles and guidelines for use in very small entities. The standard's aim was to facilitate the use of other ISO/IEC standards in very small entities through the application of life cycle profiles [36].

According to the standard, benefits that the usage of ISO/IEC 29110 provide include good internal software management processes, greater customer confidence and satisfaction, greater software product quality, increased sponsorship for process improvement and decreased development risk [2]. These benefits might also help with increased competitiveness and market share [2].

Smaller companies often have limited resources. Accordingly, the need for minimum processes and practices are supported in the scope of ISO/IEC 29110. The purpose of the Basic VSE Profile is to define a subset of processes and outcomes of ISO/IEC 12207 and products of ISO/IEC 15289 for software implementation and project management. The management and engineering guides included in the standard provide guidance for the implementation and use of a profile. The main reasons to include software implementation and project management are that the VSE core business is software development and their financial success depends on successful project completion within schedule and budget [2]. Figure 2.3 describes the ISO/IEC 29110 set of documents and positions them within the framework of the standard. Overviews and guides are published as technical reports (TR), profiles are published as international Standards (IS) [2].

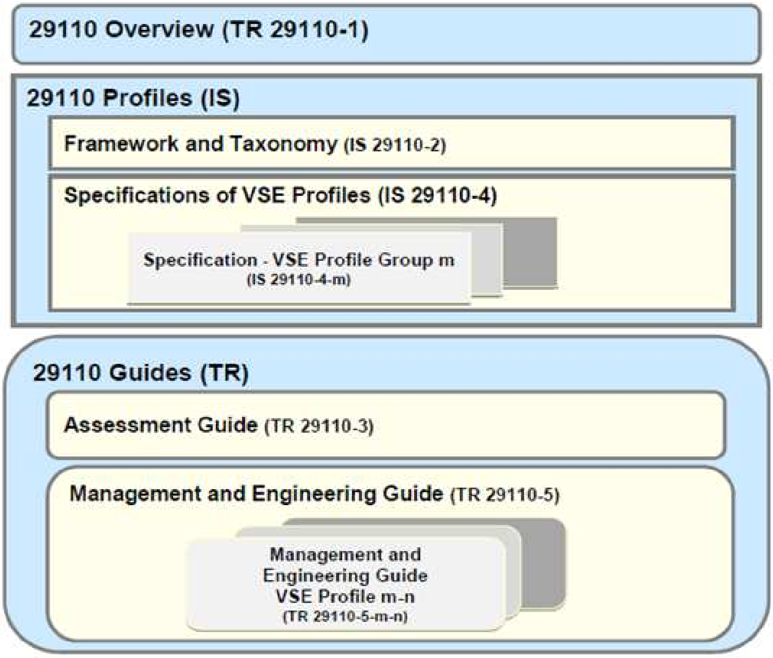

Figure 2.3: ISO/IEC 29110 Set of Documents [2].

In addition to process profiles and guidance for their usage, ISO/IEC 29110 includes process assessment framework developed to evaluate the process capability and to evaluate whether an organization achieves the targeted VSE Profile based on the evaluated capabilities for the processes. The process assessment framework is based on ISO/IEC 15504-2 [66] and uses ISO/IEC 12207 as a reference model. Figure 2.4 illustrates the relevant documents and data for a process applicable to VSE process assessment.

The ISO/IEC 29110 standard series was not yet available while initiating the research presented. Hence, the standards have not been applied in their full potential in the thesis. In addition, applying the CMMI© and/or ISO/IEC 15504 was discussed when initiating the I study of the thesis. However, the case organization stated that they did not want to apply such an exhaustive process reference model. Hence, the focus was sifted to process modeling

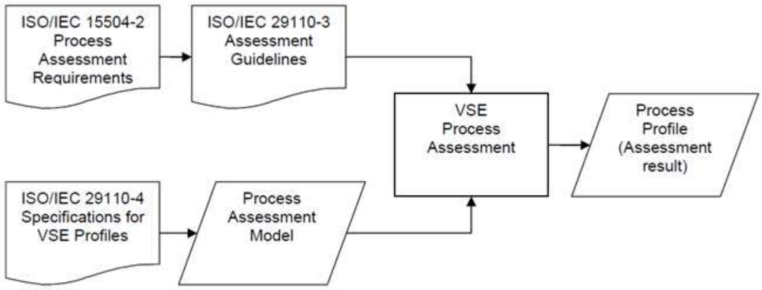

Figure 2.4: Elements of the VSE process assessment [66].

and process reference models and standards were left for future reference.

2.4 MOTIVATION FOR SPI

People factors are important to SPI success, for example Kaltio & Kinnula [67] suggest that in deploying defined software processes people factors are most important, in particular skills, motivation and time. In addition, Baddoo&Hall [34] suggest that ”SPI has a higher chance of success in companies where practitioners experience high motivation for it”. Furthermore, Baddoo&Hall [68] suggest that ”SPI may not be delivering the benefits promised because insufficient attention has been paid to the human aspects of implementing SPI”. Their view is that the non-technical, people-management factors may explain why companies are failing to achieve high process maturity [34], for example levels 4 or 5 in CMMI [17].

SPI motivators and de-motivators have been studied in different cultural contexts, for example, by Baddoo&Hall in UK [34, 68], Niazi&Ali Babar in Vietnam [69,70], and Valtanen&Sihvonen in Finland [3] (study III of the thesis). Baddoo&Hall [34] divide the motivators between different staff groups: developers, project managers and senior managers. The most important motivators among developers are researched to be visible SPI success, top-down commitment, i.e. visible senior management support for SPI, adequate resources, and bottom-up initiatives, meaning that developers have input into the design and planning of SPI. The project managers are also motivated by visible success and adequate resources. In addition, they value process ownership (practitioners own and therefore are able to change processes) and empowerment (practices within the SPI programs that empower staff to take decisions on changing processes). The most important motivational factors of senior management are visible success and meeting targets, i.e. SPI practice does not prevent the company from meeting commercial and project goals. The study by Baddoo&Hall highlights that each of the three roles require different motivators for SPI.

Niazi&Ali Babar [69] compare motivators between different company sizes. Their motivators are based on the results of a questionnaire conducted in eight Vietnamese software companies. They state that a majority of the respondents from small and medium sized companies name ”maintainable/easy processes” and ”knowledgeable team leaders” as high value motivators. Nearly 50% of them also reported ”cost beneficial” and ”job satisfaction” as high value motivators. In addition, more than 70% of SME-size companies' employees perceived ”training” as a medium value motivator.

Further, a study by Hall et al. [71] suggests that, quite opposite to the other referenced studies [3, 34, 69], ”extrinsic motivators, such as pay and recognition, appropriately fitting tasks to people and a good team infrastructure are starting to play more important roles” as software developer motivators. This finding stresses the importance of understanding and managing the non-technical factors, especially in high maturity software teams [71].

2.5 SPI IN SMALL SOFTWARE COMPANIES

Small companies have the same need to maintain and improve their competitiveness as their larger counterparts and consequently small companies also need to improve their processes [13]. This is supported by Clarke & O'Connor's study [18], where evidence is shown that increases in SPI activities increase the business success in SMEs.

Small companies find it difficult to apply standards like ISO/IEC 15504 and CMMI© in their process improvement initiatives. They often perceive that the standards are too heavy-weight for them to adopt [2, 23]. Small companies also have problems with process assessments because they usually are expensive and require significant company resources to produce meaningful results [30]. To solve these problems, several initiatives have focused on small organizations' software processes in recent years. The Software Engineering Institute initiated a CMMI© in Small Settings project in 2003 to provide approaches, tools, techniques, and guidance in small settings [72]. ISO/IEC's efforts to facilitate the adaptation of ISO/IEC standards in small companies began in 2005 when ISO/IEC 29110's development work was initiated [2].

In addition to efforts facilitating the use of standards in small companies, numerous process improvement approaches have been developed to help small companies, for example PRISMS [73], Mares [26] and Adept [25]. It is widely agreed that the special characteristics of small companies require process improvement programs to be applied in a particular way designed for small companies [74]. However, this does not mean that SPI programs intended for small companies should be scaled down versions of those applied in large companies [22].

Three small software companies' SPI initiatives have been researched by Kautz [75] with interesting results. All three companies declare that the initiative had been beneficial and worthwhile. However, the research states that there are certain conditions which might have to be fulfilled to ensure SPI success. These conditions are "external financial support, interpersonal and inter organizational networks, the assistance of the researchers as mentors for change, a tailored improvement approach, and finally the ability to use unanticipated effects". The conditions seem to correlate with general SPI success factors described in e.g. [9,35,76].

Additional SPI success factors for smaller companies can be found in Pino et al. [74] where a systematic literature review dealing with the existing SPI approaches in small and medium sized companies, along with an industrial case study, is presented. The literature review analyses 45 primary studies combined with reported practical advice to identify the success factors. The analyzed studies include improvement of various processes, such as core software development, project management, documentation, configuration management and requirement elicitation.

Pino et al.'s [74] study shows that firstly, before starting the SPI program, one should ensure that the organization is stable enough. To enable success in SPI initiatives one should initiate the improvements as soon as possible using a simple SPI model. The improvement projects should also be properly guided. Systematic following and coherent initiatives by means of specific procedures should be established. The improvements should be prioritized and the improvement points defined by the organization to allow a continuous improvement program. In addition, resistance to change can be minimized, through organizational awareness of SPI. The SPI program should be tracked and supervised by means of frequent assessments of processes, in order to evaluate the efficiency of the SPI program. In addition to assessments, measurement activities are needed to follow and ensure the success of SPI initiatives. Finally, an infrastructure to support an efficient communication between the different actors involved in the improvement is needed. Further, Pino et al.'s study highlights that process improvement activities in small companies are often varied, to include both primary software development and support services [74].

In addition, in order to ensure successful SPI, the new processes should be introduced gradually [6, 34]. Stabilizing changed processes is also important in order to make the changes permanent [77].

Numerous approaches; methods, models, and techniques have been developed to help small software organizations to model, assess, and improve their processes. Since small companies often have trouble applying standards like CMMI© and ISO/IEC 15504, there are several more lightweight improvement approaches available, e.g. Mares [26], and PEM [27], Adept [25] and RAPID [28] [78]. All these approaches aim at improving the software process via process assessments or evaluations. Additionally, there are approaches aimed to develop process guides. To point out a few, ASPE-MSC [79], and workshop oriented approach developed to the help defining electronic process guides [80]. In addition, the SPI fields' relative newcomer, lean software process development [81], has inspired the development of a new process improvement method, SPI-LEAM [82]. However, these approaches often are wide-ranging, making them difficult for a company to compare objectively and select [43, 83, 84].

Richardson [83] has recognized characteristics that an SPI approach, suitable for a small company, should have. The approach should, among other things, relate to the company's business goals, provide fast return on investment and be flexible and easy-to-use. Kautz [75] has participated in SPI initiatives in three small companies and states that the SPI assessment and improvement approach should be flexible and tailored for the company. Furthermore, Anacleto et al. [84] have identified requirements for a customized assessment method for small companies. The assessment method should have low assessment cost, reliable results, public availability, detailed descriptions and definitions, including the measurement framework, flexibility, support for the identification of risks, support for high level model, and conformity with ISO/IEC 15504. The method should be supported by a software tool covering the complete assessment process and be integrated in an assessment methodology enabling the continuous improvement of the assessment method. In addition, the assessment method should not require any specific software engineering knowledge from the company representatives.

Despite the fairly large amount of effort aimed at developing a process assessment method suitable for small software companies' needs, small companies still struggle with process assessments [30]. There are several reasons why smaller companies find process assessments difficult, such as cost, and the diversity of the processes they would like included in the assessment [23, 30]. The way some researchers have addressed these needs is by aiming at visualizing the processes and points of improvement, especially those that are informal and not documented, such as ideas, discussions, awareness and accumulated learning experienced, a category of knowledge called 'tacit knowledge' [5, 7, 85–87].

There is a growing body of research available that highlights the role and importance of tacit knowledge and small informal process improvements in small and medium-sized software companies [7, 85, 86, 88]. Clarke and O'Connor [7] state in their recent empirical examination of the extent of software process improvement in software SMEs that "the significant majority of SPI in software SMEs is minor or moderate in nature, sometimes leveraging the human capital via improvements in tacit knowledge". The results of the study show that process adaptation and optimization are important activities for software SMEs [7]. Hence, it is recommended that future research efforts focus on identifying approaches to supporting process optimization and on visualizing improvements in tacit knowledge in software SMEs [7].

2.6 PROCESS MODELING IN PROCESS IMPROVEMENT

Curtis [8] defines a model as "an abstract representation of reality that excludes much of the world's infinite detail". The purpose of a model is to reduce the complexity of a phenomenon by eliminating the detail that does not influence its relevant behavior [8].

Modeling is at the core of organizational design and information systems development [89]. Process models enable decision makers to filter out the irrelevant complexities, so that efforts can be directed toward the most important parts of the system under study [89]. The five basic uses for process models are [8]:

1. Facilitating human understanding and communication – requires that a group be able to share a common representational format.

2. Supporting process improvement – requires a basis for defining and analyzing processes.

3. Supporting process management – requires a defined process against which actual project behaviors can be compared.

4. Automating process guidance – requires automated tools for manipulating process descriptions.

5. Automating execution support – requires a computational basis for controlling behavior within an automated environment.

Process modeling has an important role in supporting process improvement activities [8]. Once a process is defined, it can be deliberately and methodically improved [8]. Process modeling helps to create process descriptions that correspond to the processes actually performed during software development or maintenance [8]. Process modeling has been used to identify shortcomings and other improvement opportunities in real world processes [5,8,87].

There are various process modeling techniques developed, for example, UML [90], Integration Definition (IDEF) techniques [89], simulation [91] and flowcharting [92]. Giaglis [89] has developed an evaluation framework for business process modeling and information system modeling techniques. The aim of the framework is to assist technique evaluation and selection depending on the characteristics of individual projects. The taxonomy applies four process modeling perspectives presented by Curtis [8]:

1. *Functional* represents what process elements are being performed, and what flows of informational entities (e.g., data, artifacts, products), are relevant to these process elements.

2. *Behavioral* represents when process elements are performed (e.g., sequencing), as well as aspects of how they are performed through feedback loops, iteration, complex decision-making conditions, entry and exit criteria, and so forth.

3. *Organizational* represents where and by whom (which agents) in the organization process elements are performed, the physical communication mechanisms used for transfer of entities, and the physical media and locations used for storing entities.

4. *Informational* represents the informational entities produced or manipulated by a process; these entities include data, artifacts, products (intermediate and end), and objects; this perspective includes both the structure of informational entities and the relationships among them.

In addition to the evaluation framework, a taxonomy of eleven modeling techniques is presented in Giaglis's study [89]. The taxonomy shows that none of the reviewed process modeling techniques satisfies all four process modeling perspectives. However, many of the presented techniques are suitable for different perspectives of process improvement. For example, functional aspects can be modeled using flowcharting or IDEF0 modeling. Behavioral side of the process can be modeled via e.g. simulation or Role Activity Diagramming (RAD) [93]. Organizational aspects can be made visible via RAD or simulation. For informational (i.e. data modeling) one can use e.g. UML.

3 Approach

The following sections present the research approach of the thesis. The research problem is presented in Section 3.1, Section 3.2 discusses the research process and Section 3.3 explains the research methods applied.

3.1 RESEARCH PROBLEM

As highlighted by the literature review in Chapter 2, despite the numerous SPI approaches and standards, the small software companies still struggle with getting their SPI initiatives going. They are in a need of a tailored improvement approach [75] which enables them to initiate the improvements as soon as possible [74]. The process improvement approach in small companies should be well focused, cost-effective and suitable for process optimization and visualizing improvements [7, 13]. Since process improvement activities in smaller software companies are often varied, to include both primary software development and support services [74], small companies require a flexible SPI approach suited to many kinds of processes. Hence, the research problem of the thesis is:

- How to practically and cost-effectively initiate process improvement in a small software company?

The research problem has been addressed via developing a suitable method for initiating SPI in small companies. In addition, research was conducted to understand what motivates a small company to implement continuous SPI. The results are presented in the five studies included in the thesis. The following section explains the research process.

3.2 RESEARCH PROCESS

The thesis presents five studies conducted to find a practical approach for initiating process improvement in small software companies. The research process is depicted in Figure 3.1. Figure 3.1 shows the five studies i.e. the five steps in the research process and how they relate. In addition, the number of case organizations involved in each study are shown in the figure.

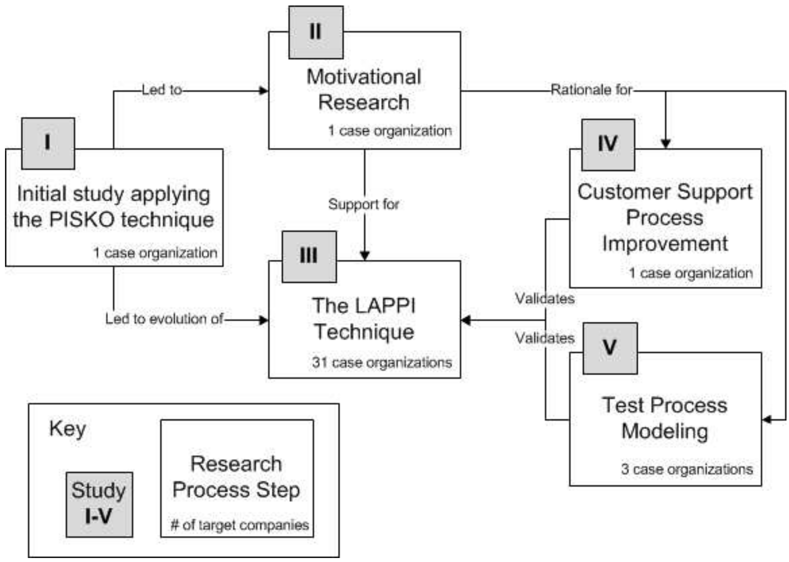

Figure 3.1: The research process.

3.2.1 Initial Study

The starting point of the research was a small software company participating a research project where the aim was to practically improve the processes of SME-sized software companies with limited knowledge of SPI. The first step of any such project is to make the current state of the target company's software process visible [8,9].

One of the basic problems with small companies SPI, in addition to the often limited resources, is a difficulty with initiating

SPI [13]. In general, the common way to initiate SPI is to apply an SPI standard e.g. CMMI or ISO/IEC 15504, and start the improvement project by implementing a related process assessment [9, 30]. However, small companies often have problems with the SPI standards [2] and process assessments because the costs and significant resources required to produce meaningful results [30]. Hence, this thesis presents an optional approach; process modeling as the first step of SPI.

A process modeling technique named PISKO was published in 2002 [87]. PISKO has been applied in several studies, see e.g. [87, 88, 94]. PISKO has been shown to be suitable for modeling the current process, identifying the points of improvement and identifying problem points in the process. In addition, applying PISKO does not require any previous knowledge of SPI from the target company. Further, PISKO makes processes and its problem points visible with a modest amount of resources.

The I study of the thesis, PISKO was applied to initiate SPI in our target company. The results were promising, the current process was made visible, the problem points and points of improvement were identified.

3.2.2 Motivational Research

After implementing the initial study applying the PISKO technique, the target company's employees' motivation for SPI was high. Since motivation appears to be one of the main enablers of successful SPI [34], it was important to understand what motivated the employees. Hence, a motivational study was conducted as the II study of the thesis. As a result, a better understanding of the factors creating motivation for SPI was accomplished. Further, the motivational study helped to better prepare for the next steps in the target companys SPI initiative, the customer support process improvement presented in study IV and test process improvement presented in study V.

3.2.3 The LAPPI Technique

Despite the promising results in study I, it was presumed that a deeper understanding of the process could still be reached via process modeling. For example, when problems with roles and responsibilities in the process were revealed, PISKO was not able to illustrate them properly making it hard for the researchers to understand the big picture. This notion had also been made in other studies piloting PISKO. Hence, the improvement of the technique had been initiated soon after publishing the PISKO technique in [87]. Trials, where role and information flow modeling was combined to the PISKO technique had been performed with promising results. After study I, the author of the thesis took an active role in improving the PISKO technique. At this point the technique was at an evolution phase where the original process modeling sessions were supported by role and information flow modeling. The process modeling had been divided in two workshops. The information flow modeling was performed first and process modeling as the second step. The next step in evolving the technique was to create documentation templates to support the modeling work. This was done as part of the research presented. However, the document templates did not provide the expected results and their usage was soon abandoned. As a result of five evolution phases, see [5], the technique was developed into its current form and named LAPPI: A light-weight technique to practical process modeling and improvement target identification. Study III of the thesis presents the LAPPI technique.

3.2.4 Customer Support Process Improvement and Test Process Modeling

The LAPPI technique was validated as part of this thesis. In addition to the validation presented in study III, two validating studies were conducted. The IVth study of the thesis was implemented in the same small company where study I was conducted. Study IV focused in improving the customer support process of the company applying LAPPI. The Vth study included three software companies. Here, LAPPI was validated via modeling the testing processes. The five studies presented together report a practical approach for initiating process improvement in small software companies. Further, the results of the motivational research conducted help small software companies towards continuous SPI.

3.3 RESEARCH METHOD

Different research methods were applied in the five studies that frame the thesis. In the I study, action research [95] was used. The II study was conducted as a form of a case study [96]. In III study, a non-formal variation of constructive research [97] was applied to develop the LAPPI technique. Study IV was conducted using action research and V followed the case study process. In the following sections, the research methods and how they were applied in the studies is discussed in more detail.

3.3.1 Action Research

According to Greenwood & Levin [98] ”Action research is a social research carried out by a team that encompasses a professional action researcher and the members of an organization, community, or network (”stakeholders”) who are seeking to improve the participants' situation.” Further, action research is stated to simultaneously assist in practical problem solving and expand scientific knowledge [95].

The major strength of action research is the practical and deep understanding the researcher obtains [99]. In addition, action research process is very similar to the IDEAL SPI program model [32]. Its weakness is the potential lack of objectivity on the part of the researchers when they attempt to secure a successful outcome for their client organization [99].

Action research is conducted in five iterative phases [95]:

1. diagnosing,

2. action planning,
3. action taking,
4. evaluating and
5. specifying learning

Action research is a practical research method helping the researchers try out their theories with practitioners [100]. Its popularity is increasing in information systems' research [95] and in Conradi & Fugetta's paper of improving software process improvement [29] it is suggested, together with multidisciplinary teams, as a partial solution for the fundamental method problem of how to influence and study social organizations. In addition, Sjøberg et al. [101] conclude in their paper about the future of empirical methods in software engineering research that action research provides the most realistic research setting among empirical research methods. This is due to the fact that the setting of the study is the same one in which the results will be applied apart from the presence of the researchers [101].

In this thesis action research was applied when the active participation of the researchers was needed to implement the study. Hence, action research was the method of choice in studies I [88] and IV [31]. Each of these studies involved process modeling where the researchers acted in leading roles of the modeling workshops. This was necessary in order to test how the outside support would affect the SPI initiatives.

In the studies, the five phases of action research were implemented to model the processes at hand. In the I study the target was software engineering process and in study IV, the customer support process. In both studies diagnosing and action planning phases were conducted while preparing for the process modeling workshops. Action taking was the process modeling itself. Evaluating was conducted via the identifying the points of improvement and specifying learning phase was where the produced process model and conducted improvements were reflected.

3.3.2 Case Study

A case study is an empirical research method commonly applied in software engineering research [101, 102]. Case studies are best suitable in situations where 1) a how or why question is being asked, 2) the focus is on a contemporary set of events, 3) over which the investigator has little or no control [96]. Case studies are also useful in answering a "which is better" question [103]. In software engineering case studies should be used not only to decided which is better but also to evaluate the differences between, for example, two design methods [104].

According to Yin [96], a complete case study protocol includes the following: (a) The procedures for contacting key informants and making field work arrangements; (b) explicit language and reminders for implementing and enforcing the rules for protecting human subjects; (c) a detailed line of questions, or a mental agenda to be addressed through out the data collection, including suggestions about the relevant sources of data; and (d) a preliminary outline for the final case study report.

Quality of case studies can be assessed via four basic social science tests [96]. The tests can be passed via applying suitable tactics as described below [96]:

1. *Construct validity* can be ensured applying multiple sources of evidence, establishing a chain of evidence and having key informants review draft case study reports in data collection and composition phases.

2. *Internal validity* can be ensured via doing pattern matching and explanation building, addressing rival explanations and using logic models in data analysis phase.

3. *External validity* can be ensured via using theory in single-case studies and using replication logic in multiple case studies in research design phase.

4. *Reliability validity* can be ensured via using case study protocol and developing case study database in data collection phase.

Case study designs can be single-case or multiple-case studies, and they can involve a single unit (holistic) or multiple units (embedded) of analysis [96]. In software engineering, case studies are important because they are needed for the industrial evaluation of developed methods and tools [101]. Applying case studies, researchers can avoid the scale-up problems that are often associated with experiments [101]. Internal validity can be a problem in software engineering research case studies. There are three ways of designing a software engineering case study to ensure internal validity and avoid bias [103]: results can be compared with a company baseline, with a sister project, or components within a project can be compared.

According to Runeson et al. [102] many research questions in software engineering are suitable for case study research. This is mainly because of software engineering's multidisciplinary nature. Research on software engineering is most often aimed at investigating how software development, operation and maintenance are conducted by software engineers and other stakeholders under different conditions. Runeson et al. also note that "software development is carried out by individuals, groups and organizations, and social and political questions are of importance for this development."

In this thesis two case studies were conducted. The first one was the motivational research implemented in study II. This study was a single-case study involving a single unit of analysis; one small software company. Here the data collection was implemented via interviews [105] and a survey [106] which are commonly used data collection instruments in case studies [96]. The second case study was study V. There, the LAPPI technique was validated via a multiple-case study where the testing processes of three software companies were modeled and their points of improvement identified. For the Vth study, action research would have also been a good choice. However, case study was chosen because it would make it easier to compare the results of each company.

3.3.3 Constructive Research

Constructive research aims at producing novel solutions to both practical and theoretical problems [97]. The constructive approach is based on a design of constructs or constructions of a solution. Constructive research is suggested to be applied to producing solutions to explicit problems: a change process and/or something which is profoundly different from existing concepts, something that produces new reality and its usability can be demonstrated through the implementation of solutions [107]. The features of the constructive research approach are presented in Figure 3.2.

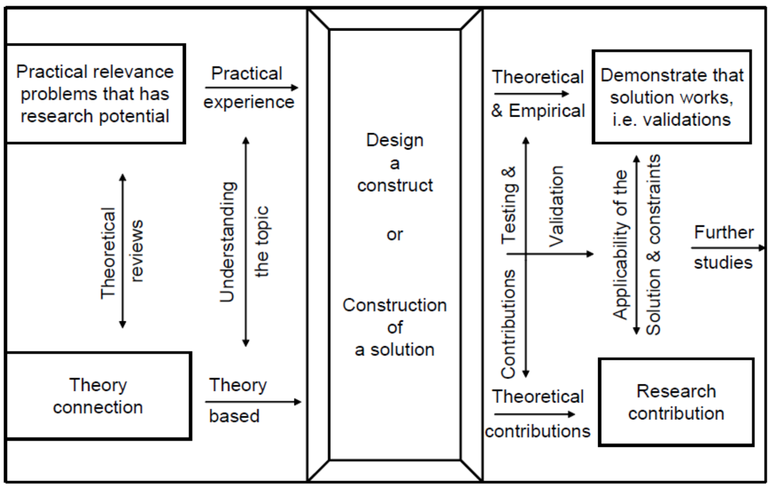

Figure 3.2: The features of the constructive research approach [97]

.

Despite being a less discussed research method in software engineering research, constructive research is not a new approach. Oyegoke [97] reports that there are several applied constructive studies in technical sciences (new product development), clinical medicine (creation of a new treatment), finance (option pricing), philosophy (creating artificial language), management accounting (a new budgeting system) and operations research.

The possible limitations of constructive research are that a solution (the construct) that is considered technically adequate is not necessarily guaranteed to work in practice [107]. These problems can be tackled by linking an early problem definition stage to both practical and theoretical problems, and the involvement of the organizations applying the construct in designing and implementing the solution [97]. Furthermore, another possible limitation is the scientific prowess of the constructive research approach. However, constructive research passes the objectiveness and criticalness of applied research [97].

A non-formal variation of constructive research was applied in developing the LAPPI technique. The construct of the study was the LAPPI technique itself that was evolved via five evolution phases explained in [5]. The construct was shown to work in practice via 42 validating studies.

4 Validating the Approach

In this Chapter the five separate studies and their results are presented. The five studies are divided into approach development studies, presented in Section 4.1, and approach validation studies, in Section 4.2. The aim of the studies was to find an practical approach to initiating process improvement in small software companies. The aim is met by offering a validated process modeling technique, LAPPI, suitable for this task. In addition, motivation monitoring, consisting of motivational interviews and a survey, is proposed to be used to support SPI initiation and to enable continuous SPI. The main contribution of the thesis is presented in Section 4.3.

4.1 APPROACH DEVELOPMENT STUDIES

The studies related to developing the approach for small companies to initiate process improvement are presented in this Section.

4.1.1 Study I: Initial Study

Research Paper: Anu Valtanen and Jarmo J. Ahonen: Big Improvements With Small Changes: Improving the Processes of a Small Software Company [88]

Summary: Study I describes the first steps of SPI in a small software company. When initiating the improvement efforts, the processes were at zero level, the employees were not even quite sure whether they used such things. However, the whole company was eager to improve their operations. The first SPI initiative was executed by modeling the software process using the PISKO technique [87]. The problem points were identified, the software process streamlined and improvements implemented.

The results: As a result of the process modeling and improvement efforts process culture and basis for the company's SPI program were created. Further, high motivation for SPI was gained. The results indicate that one practical way to initiate process improvement in a small software company is applying process modeling. Process modelings enables making the process and its problem points visible with a modest amount of resources. To improve the process, it is suggested optimize it; to clarify the process by omitting unnecessary phases and assigning clear pre- and post- conditions for process phases. Further, the the process and roles and responsibilities related to it should be made clear for all the stakeholders. Nominating a person in charge of the processes is important. Further, a need for improving the PISKO technique became apparent and an active role in the technique's improvement efforts was taken by the author.

4.1.2 Study II: Motivational Research

Research Paper: Anu Valtanen and Hanna-Miina Sihvonen: Employees' Motivation for SPI: Case Study in a Small Finnish Software Company [3]

Summary: High motivation for SPI among the case organization was registered during the improvement initiative of the study I. It was understood that with a small company's limited resources, motivation of the employees would be essential in order to enable continuous process improvement. To find out how to maintain the motivation it was decided to conduct motivational research. A series of interviews and a motivation survey were conducted in the case organization. The interviews were designed to measure the overall SPI experience of the employees. The survey on its part was based on previous motivation factors presented by Baddoo & Hall [34].
The results: The motivational factors of the employees and the CEO of the case organization were quite different in nature. The CEO

named meeting targets, cost beneficiality and visible success as his main motivators. The employees consider that job satisfaction, autonomy and standardization are the most important motivators. Perhaps not suprisingly, the CEO named productivity related issues to have the most positive impact on his motivation. Meanwhile, the employees most appreciate factors that make their jobs easier. In addition to revealing the factors that motivate the employees and the management, motivation survey and the interviews proved to be useful tools in planning the future SPI strategy.

4.1.3 Study III: The LAPPI Technique

Research Paper: Anu Raninen, Jarmo J. Ahonen, Hanna-Miina Sihvonen, Paula Savolainen and Sarah Beecham: LAPPI: A Lightweight Technique to Practical Process Modeling and Improvement Target Identification [5]

Summary: The LAPPI technique and its evolution process is described in detail in study III. LAPPI is a practical and lightweight modeling technique designed to suite all kinds of process improvement needs. It is especially useful in the process improvement initiation phase.
The results: The LAPPI process modeling technique, that pays particular attention to ease of use, speed, and low utilization cost, has been developed in order to facilitate SPI initiation in SME-sized software companies. LAPPI is an evolved version of the PISKO modeling technique, described in [87], that allows for role and information flow modeling. The result of the LAPPI technique is a process overview and improvement suggestions generated via process modeling workshops. The LAPPI technique has been widely tested in software industry. In study III, we present 42 modeling cases conducted in 31 companies where LAPPI has been applied and validated.

4.2 APPROACH VALIDATION STUDIES

The studies related to validating the LAPPI technique are presented in this Section.

4.2.1 Study IV: Customer Support Process Improvement

Research Paper: Anu Raninen, Helena Merikoski, Jarmo J. Ahonen and Sarah Beecham: Applying Software Process Modeling to Improve Customer Support Processes [31]

Summary: Customer support process was a natural improvement target after the software engineering process in our case organization from study I. Customer support has a significant role in organizations' operations, because they offer product software: products that are not developed for one specific customer, but for an entire market [108]. The customer support improvement project was executed using the LAPPI technique. The success of the improvement efforts was measured via a customer satisfaction survey administered in two subsequent years. One survey before applying LAPPI, and another a year later, to measure the success of the improvement efforts.
The results: Applying the LAPPI technique highlighted that the organisation had problems with their customer support services. Consequently the customer support process was streamlined. The results of the customer satisfaction surveys were encouraging: customer satisfaction improved in several areas, with significant improvement in support request response times, and in customer support request resolution.

4.2.2 Study V: Test Process Modeling

Research Paper: Tanja Toroi, Anu Raninen and Lauri Väätäinen: Identifying Process Improvement Targets in Test Processes: A Case Study [33]

Summary: The V study presents the results of case studies conducted in three software companies in order to identify the improvement targets in their test processes. The testing processes were modeled using the LAPPI technique and based on the modelings, improvement suggestions were given.
The results: The most significant problems identified in this study were the low level of unit testing, unknown repercussions of code changes, and missing exit criteria for testing. In addition, a problem that has not been frequently reported in previous research was identified. It appears that software companies find it difficult to integrate test automation with their manual test processes. Making the recurring test related problems known among the software industry helps to justify the need for test process improvement and standardization of test processes.

4.3 RESULTS

Previous SPI research suggests that in order to initiate SPI in a small company, a light-weight, easy to apply approach that produces visible results quickly is needed. The results of the research presented in this thesis support these notions. In addition, the motivation of the employees of the company aiming to improve is highlighted. In this section the main contribution of the thesis is presented. Section 4.3.1 gives an overview of LAPPI, a technique for SPI initiation. In section 4.3.2 motivational monitoring, a way to gather information on how SPI initiatives should be initiated and implemented in small companies is discussed.

4.3.1 The LAPPI technique

LAPPI (A Light-weight Technique to Practical Process Modeling and Improvement Target Identification) [5] provides an easy to use, lightweight tool for process modeling and improvement target identification. The method used in the development of LAPPI is a nonformal variation of constructive research [97].

The LAPPI technique makes the process visible and its problem points visible through process modeling [8]. Using LAPPI does not exclude the usage of reference models like CMMI or ISO/IEC 15504 and associated process assessment approaches. However, LAPPI is suggested to be used instead of standards-based process assessment to initiate SPI. In relation to the Initiating-Diagnosing-Establishing-Acting-Leveraging (IDEAL) model [109], which can be used as basis for an SPI program, LAPPI is most beneficial in the diagnosing phase of SPI. The result of applying the LAPPI technique is an understanding of the workings of the current processes and the organizational interactions and a process description baseline.

The LAPPI technique consists of 13 steps presented in Figure 4.1. The key of the technique are the two modeling sessions. In the first one, the roles and information flows are modeled. In the second session the process is made visible. In both sessions a wall-chart technique is applied [87]. LAPPI involves two teams; ’leading team’ and ’customer organization’. The leading team is responsible for chairing the modeling workshops and creating the documentation that is the result of LAPPI. The process modeling workshop participants form the customer organization. The LAPPI technique can be applied with or without external help. With external help, the leading team comes from outside the organization. As a result, the LAPPI technique produces a process overview and improvement suggestions. The process overview consists of a process model and diagrams/tables describing the process phases and participating roles.

LAPPI has been developed in collaboration between academia and industry. Validation of the technique in 42 studies conducted in 31 companies presented in Raninen et al. [5] indicates that the technique is suitable for modeling the current process and identifying the points of improvement in the process. Practical experience shows that LAPPI provides a cost-effective technique for process modeling and improvement target identification especially in small and medium-sized enterprises.

The LAPPI technique has been validated through the 42 studies

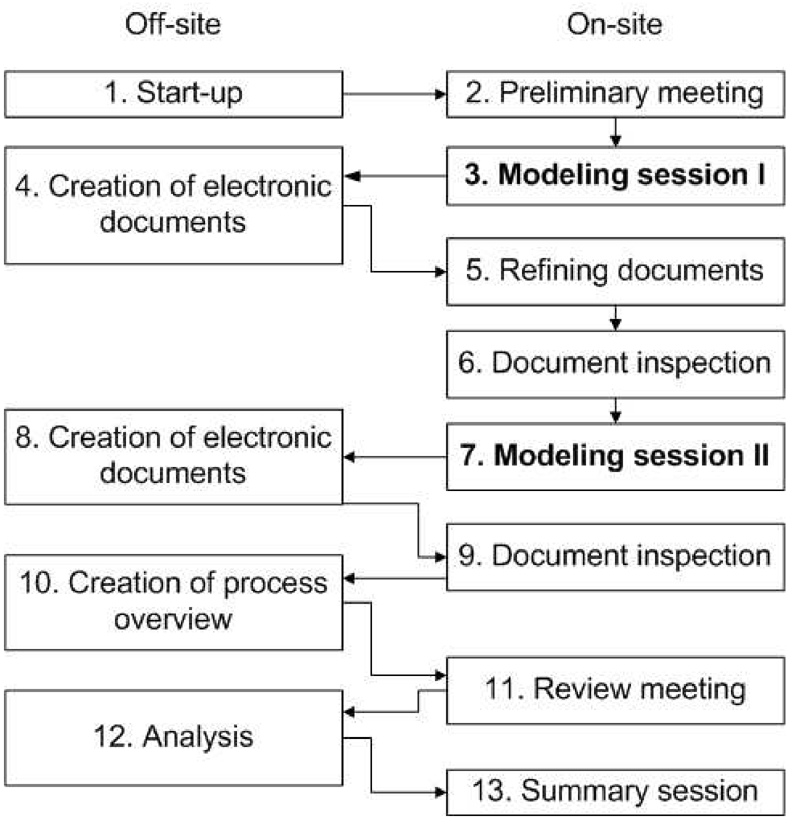

Figure 4.1: The LAPPI technique's 13 steps [5].

reported in study III [5]. In addition, four validating case studies are presented in this thesis. These four studies apply the most up-to-date version of the LAPPI technique. In study IV the technique is applied to model a customer support process. In study V, LAPPI is used to model testing processes of three companies. In both cases the modeling was conducted with external help. The author of the thesis acted as the chairman of the modeling workshops i.e. the head of the leading team.

In study IV, applying the LAPPI technique made the case organization's problems with their customer support services visible. Consequently the customer services process was streamlined. We validated our approach by examining responses to customer satisfaction questionnaires administered prior to and post our LAPPI intervention. Results were encouraging: customer satisfaction improved in several areas, with significant improvement in customer response times, and in customer support request resolution [31]. Customer support has a significant role in the case organizations' operations, because they offer product software [108]. Applying an SPI approach to customer support process improvement turned out to be beneficial. In this case minimal amount of resources, 101 man-hours (65 hours from target company + 36 hours of researcher time), was required to provide remarkable improvements. As a result, it appears that improving customer support and integrating it tightly with the software engineering process helps to improve customer satisfaction.

In study V, test process improvement projects were initiated using LAPPI in three case organizations. The problems identified applying LAPPI were fairly similar to those reported in earlier test process improvement related studies. As a result of study V, process modeling can be recommended as means to get the test process improvement projects going. A light-weight way of making the problems of test processes visible appears to help to underline the importance of test process improvement. Also, seeing the concrete problems might convince the companies to exploit the test related standards and test improvement approaches to support the process improvement.

Further, the results of the study V support that the LAPPI technique is a simple and cost-effective tool to identify process problems with small amount of resources. At the minimum, it took only 83 man-hours to provide a case organization with a process overview and identified process problems. In addition, improvement suggestions are easy to provide based on the detailed modeling results.

4.3.2 Motivation Monitoring

In small software companies SPI initiatives, the motivation of the employees plays on important role [3]. The human resources represent a crucial part of companies process infrastructure [67]. Hence, their motivation for SPI is an important factor for successful improvement initiatives and implementation [3]. However, previous studies of SPI motivation have concentrated on exploring motivation mainly in large and SME sized software companies, and not emphasized the small and very small software companies. Study II provides a more comprehensive perspective of small software companies employees motivation and point of views on SPI [3].

Data collection method of study II consisted of two basic instruments: individual interviews and motivation survey designed by Baddoo et al. [34]. The instrument is described in detail in study II [3]. According to the motivation survey, the most important factors having strong positive impact on motivation were top-down commitment, shared best practices, resources and bottom-up initiatives. The most important motivators having a positive impact were autonomy, feedback and justifiable benefits. The factors the employees themselves chose to be the most important, were job satisfaction, standardization, and autonomy.

The results of study II suggest that motivation monitoring in form of the motivation survey and the interviews are useful tools in planning the SPI strategy. A lot of valuable information was discovered for planning and implementing the next steps of SPI. Hence, when initiating SPI or planning future SPI cycles in small software companies, it could be useful to carry out a motivation monitoring before SPI actions or latest, in the early stages of the SPI initiative. As a result, the companies have a better understanding of how to involve and interact with the employees or management.

For example, based on study II, we can conclude that in the future SPI initiatives, we should aim at increasing job satisfaction. First step should perhaps be coming up with a way to measure the satisfaction. In addition, the employees should be given freedom to plan and implement SPI themselves where possible, since they are motivated by autonomy. Further, we can state that the employees are not afraid of standardized ways of work which should ease the improvement whereas process formalization might be needed.

5 Discussion

In this thesis, an optional approach for initiating SPI in small companies is presented. Here, process modeling using the LAPPI technique [5] is suggested as the first step of SPI. Further, motivation monitoring [3] is recommended to be applied in order to support enabling continuous improvement in small software companies. The proposed approaches' position in relation to the IDEAL model [109], which can be used as a roadmap for an SPI initiative, is presented in Figure 5.1. LAPPI and motivation monitoring are shown to be beneficial in the initiating stage of SPI initiatives, see studies I-V.

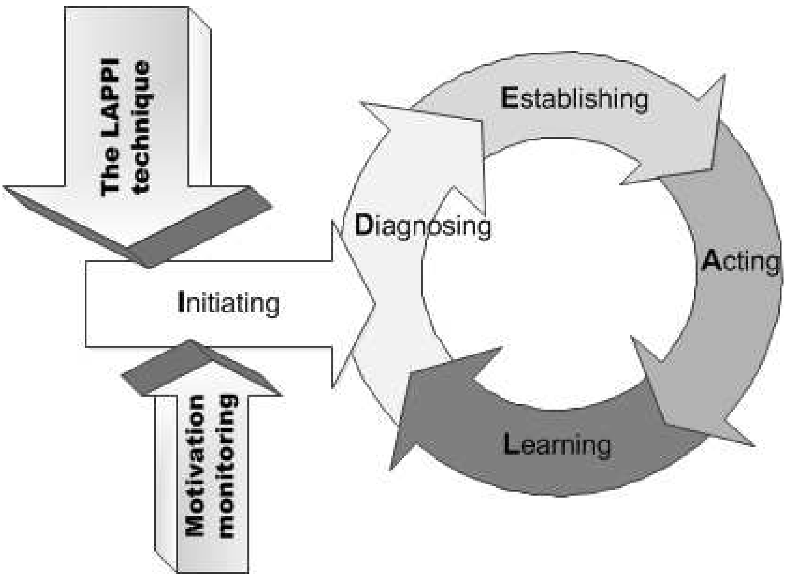

Figure 5.1: The LAPPI technique and motivation monitoring in relation to the IDEAL model.

The majority of software companies are small [11,36]. The small companies need to improve their operations to maintain their competitiveness [13]. To attain better product quality through improved processes SPI has proved to be a beneficial solution [19, 20, 44–46]. When a process is fit for its purpose, it produces satisfactory products [6]. However, small companies often have problems with initiating SPI. They say that the standards and approaches developed for SPI adoption are too heavyweight and hard to implement [2, 13, 23]. Further, a recent study by Clarke and O'Connor [7] shows that smaller software companies prefer to make small adjustments to the software process on a regular basis rather than implement a few major SPI initiatives.

For successful SPI initiation in small companies, a light-weight approach that can produce visible results quickly is needed [83]. Approaches aiming to fulfill this requirement have been developed in previous research [25, 26]. However, these approaches are usually based on process assessment. Process assessments measure the fitness of the software process in relation to a standardized process reference model, for example one presented in CMMI or ISO/IEC 15504. The idea is that via light-weight process assessments small companies can more easily adopt the reference models and apply them as a basis for their SPI initiatives. However, small companies find the process assessments difficult to implement [23, 30]. In addition, small informal process improvements and tacit knowledge may not be easily detectable using traditional process assessment approaches [7].

The proposed approach for SPI initiation does not exclude the usage of process reference models and process assessment in the later stages of SPI. Applying the LAPPI technique enables the visualization of tacit knowledge, process optimization and implementing small informal process improvements. Motivation monitoring supports spreading tacit knowledge and helps the companies to better understand how they could support their employees with striving for continuous improvement. The LAPPI technique and motivation monitoring are suitable to be used to initiate small informal process improvement as well as larger SPI initiatives supported by SPI models and standards.

The suggested approach for initiating improvement is discussed in the following sections. First, the LAPPI technique and its applicability are evaluated in section 5.1. Second, the motivation monitoring is reflected in section 5.2. Third, limitations of the study and future research are discussed in section 5.3.

5.1 THE LAPPI TECHNIQUE

As shown by the validating presented in studies III-V [5,31,33], the LAPPI technique enables small companies to initiate SPI by making processes and their problem points visible via process modeling [8] instead of standard based process assessment. A process model enables decision makers to filter out the irrelevant complexities, so that improvement efforts can be directed toward the most important parts of the process [89]. The result of the LAPPI technique is a process overview and improvement suggestions generated via the process modeling workshops. LAPPI is practical, easy to apply, lightweight and publicly available. Using LAPPI, the processes are modeled in a practical and cost-effective way. The technique involves the employees of the target company and makes the resistance to change somewhat smaller [5, 88]. Despite the fact that LAPPI does not make the usage of standards like ISO/IEC 15504 or CMMI© mandatory, they can be, and are often beneficial, used as reference models during the improvement projects.

Previous research has shown that, in small companies, SPI should be initiated using a simple, flexible and cost-effective approach [74, 83]. The improvements to be implemented should be prioritized and points of improvement defined to allow a continuous improvement program [74]. LAPPI has been shown to be a simple and flexible technique that produces visible results quickly [5]. However, it does not provide a mechanism for prioritizing the improvements.

Further, the industry people, who were asked to evaluate LAPPI in study III [5], state that the technique makes them feel part of the improvement initiative. Employee participation has shown to be important, people are more likely to support what they have participated in creating [80]. Increased participation leads to better solutions and enhanced organizational problem solving capability [110]. Further, the evaluation of the technique shows that LAPPI is good for motivation, one of the most important SPI enablers [34,35]. The LAPPI technique is shown to be suitable for process optimization which is one of the main ways how SME-sized companies improve their processes [7,88]. In addition, LAPPI enables the tacit knowledge to be used as an input for the SPI initiative via making processes, roles and information flows visible in a way other methods cannot. The visualization of tacit knowledge is supported by separate modeling session for roles and information flows related to the process. In addition, LAPPI stimulates discussion and makes people feel invested in SPI [5]. A more detailed discussion of the strengths and weaknesses of the LAPPI technique is presented in study III [5].

5.2 MOTIVATION MONITORING

Management and employee motivation and commitment supports successful process improvement [34,35]. Further, to attain the best results, SPI must be a continuous effort [9,76]. Hence, it is important to understand what motivates a small company to implement continuous SPI. In the case organization of study II the motivation for SPI was high from day one. It was understood that this high level of motivation should be maintained in order to make the SPI initiatives continuous. Motivational research, including interviews and a survey about motivational factors [34], was conducted in order to find what motivates the employees for SPI and how to maintain the motivation.

The motivational research revealed that employees find it motivating to perform their different roles in the company especially when these roles are well defined [3]. This was evident because autonomy, communication, process ownership, shared best practices, standardization, and training were seen as highly motivating factors. In addition, the results indicated that employees are motivated by surprisingly regulative and perhaps even restrictive work practices. The employees stated that they gain job satisfaction from producing and following a good quality process. In addition, they hope that SPI will make them work in a more standardized way. These motivators are similar to those presented in earlier studies, see [34] and [69].

The management's motivators were also researched. The target company's CEO's motivators were quite similar to those detected in previous research [34]. He stated that it would be motivating to meet targets and have an improved cost/revenue ratio through SPI. In addition, visible success motivated him. Hence, it is clear that managers are motivated differently to other employees.

The motivational interviews and the questionnaire offered valuable input for planning and implementing the future SPI initiatives, in addition to measuring the motivational factors of the employees and the management. Hence, motivation monitoring, consisting of the interviews and the survey, is proposed to be used in software companies before initiating SPI or by the latest, in the early stages of the SPI initiative. The research presented indicates that motivational monitoring helps in ensuring continuous SPI.

5.3 LIMITATIONS AND FUTURE RESEARCH

The limitations of the research presented and implications to future research are covered in this Section.

5.3.1 Threats to Construct Validity

Construct validity is defined as "identifying correct operational measures for the concepts being studied" [96]. Hence, to show construct validity, a demonstration that a test is measuring what it claims to be measuring is needed.

Despite the LAPPI technique being validated extensively in studies III, IV and V, the results are mostly based on the opinions of the case organizations' employees. This is because formal measurement of SPI success is often problematic [111]. In addition to the customer satisfaction survey presented in paper IV, there are no other quantitative measures in place. The measurement of the SPI initiation success is based on the motivational interviews presented in study II and other discussions with the case organizations. This issue is a complex one. On the other hand, in small companies, the opinions of the employees appear to be one of the most relevant measures in improvement [7]. However, it would be beneficial to have quantitative data to support the applicability of LAPPI and motivation monitoring.

Further, the LAPPI technique was developed using a non-formal variation of constructive research. The possible limitations of constructive research are that a solution (the construct) that is considered technically adequate is not necessarily guaranteed to work in practice [107]. The limitations of the constructive research method in study III are discussed in [5].

In addition, the action research collaborative framework diminishes the researcher’s ability to control the process and the outcomes of the research [95]. Action research, applied in studies I and IV, looks a lot like consultancy [95]. This may confuse the case organizations as they may expect consultancy-type performance from the researchers, which is also a threat to construct validity. The author has been conscious of this risk from the beginning and has made sure to carefully explain the research settings to the target company.

5.3.2 Threats to Internal Validity

Internal validity refers to causal relationships between different interpretations of research results [96]. For example, how well a piece of research allows you to choose among alternate explanations of something.

An internal validity threat to be taken into account in the thesis is that the observed improvements cannot be claimed to be unambiguously due to the LAPPI intervention in study IV [31]. Despite the customer satisfaction surveys showing a trend of improvement, there may be various other reasons for customers reporting higher levels of confidence in customer support in the company.

5.3.3 Threats to External Validity

External validity deals with the problem of knowing whether a study's findings are generalizable to other contexts [96]. Generalization of the results is one of the biggest limitations of case studies [96]. Generalizing the result from few cases may be dangerous.

In studies III, IV and V, we show that the LAPPI technique is applicable in identifying the problem points of different kinds of organizations and processes. However, LAPPI has only been applied in Finnish companies which is a limitation to external validity. In addition, the motivation monitoring in study II is only shown to be beneficial in the one company in which our study was conducted and, as such, it needs further validation. Therefore, although 46 case studies were conducted, we cannot generalize the results outside the given context of SMEs delivering product software in Finland.

5.3.4 Threats to Reliability

Reliability threats are to do with other researchers' ability to repeat the conducted studies [96]. The LAPPI technique is described in detail in Raninen et al. [5] and should be repeatable. Further, motivational research presented in study II [3] is a replication of Baddoo&Hall's study [34] and should be replicable also in other contexts.

However, what cannot be unambiguously claimed, is that the results of applying LAPPI would be exactly same if implemented by other researchers. One of LAPPI's limitations is its potential dependability on the chairman's skills. The chairman leads the modeling sessions. In addition, wrong people participating the modeling may make the effort void. These limitations are discussed in [5].

5.3.5 Future Research

An alternative approach for initiating SPI in small software companies is proposed in this thesis. The approach: the LAPPI process modeling technique and motivation monitoring, are suggested to be applied in the initiation phase of process improvement projects to get the SPI initiatives going. The validation of LAPPI has been presented in studies III, IV and V. The motivation monitoring is applied in study II. Despite the fact that the approach has been shown to work in the presented studies, further research is needed to strengthen the validation of the approach.

Further research is needed to study whether the approach can be applied in other contexts, preferably outside of Finland. The research presented could be extended by applying the LAPPI technique in other countries and cultural environments. In addition, the motivation monitoring could be applied in additional companies in Finland and elsewhere. Motivation survey presented in study II is a replication of Baddoo&Hall [34], used also by Niazi&Ali Babar in [70]. However, in those studies the focus was mainly on researching what motivates the SPI practitioners. In this thesis, a new way to apply the motivational survey results is proposed. In addition, the motivational interviews conducted in study II [3] is a new instrument. The approach suggested here is to conduct the motivation interviews and survey before or in the beginning of an SPI initiative. As a result, the companies can form a better understanding of how to involve and interact the employees and/or management in the SPI initiatives. This approach needs further validation. Motivation monitoring needs to be applied in other companies to see whether it brings additional value in more than one context.

In addition, the quantitative measurement of the SPI initiation approach's effectiveness would be an interesting research topic, as suggested in Section 5.3.1. It would be beneficial to have more quantitative data to support the applicability of LAPPI and motivation monitoring. However, there are not that many unambiguous measures available to demonstrate the success of SPI initiatives in general [111]. Hence, further research is still needed to first, develop reliable measures for the success of SPI initiation and second, to collect data in software companies. Results from replicating the approach in different contexts will indicate whether the results can be reliably generalized.

6 Conclusion

Small companies have trouble applying the existing software process improvement approaches and standards. With their limited resources they especially struggle with initiating SPI. It would appear that the traditional SPI approach (process improvement supported by process reference models and process assessment) is not necessarily the best possible solution for SCs. In this thesis an optional approach is presented.

Previous research has shown that small software companies prefer to make small adjustments to the software process on a regular basis. In SCs SPI, process optimization is an important activity. Further, smaller companies SPI is characterized by the need to be focused and tacit knowledge-led where informal improvements in the form of ideas, discussions, awareness and accumulated learning experienced are important [7].

The main contribution of the thesis is the LAPPI process modeling technique. As shown in study III, LAPPI enables making processes and their problem points visible in a practical and cost-effective way. The LAPPI technique is designed to be applied as the first step of SPI in small software companies. LAPPI supports small adjustments and tacit knowledge-led SPI by visualizing the tacit knowledge and points of improvement. For example, in study IV, the majority of identified problems were addressed through improvements in communication and making tacit knowledge explicit and defining employee roles. The support is provided via highlighting the roles and information flows related to the process. A separate modeling session specifically for role and information flow modeling is an important part of the LAPPI technique [5].

Further, to complement the LAPPI technique, motivation monitoring is proposed to support initiating SPI and enabling continuous improvement. Motivation monitoring provides an improved understanding on how to involve the employees in SPI initiatives.

The results of study II strengthen the results of previous studies [34, 70], where it is shown that managers and employees are motivated differently. In our study the CEO named productivity related issues to have the most positive impact on his motivation. Meanwhile, among the employees, factors that increase job satisfaction and facilitate executing the processes were seen as the most significant. Hence, it can be concluded that, when implementing SPI programs, managers cannot assume that their employees will be motivated in the same way as they are. Further, the research presented highlights that a motivated workforce will ease the implementation and support of continuous SPI [3].

The suggested SPI initiation approach, LAPPI complemented by motivation monitoring, helps a company to understand their current processes, the problematic areas of the process, and organizational interactions. In addition, the SPI initiation approach results in creating a process description baseline. This approach supports the small companies with process optimization and implementing small adjustments to the software process on a regular basis. However, applying the LAPPI technique does not exclude the usage of process reference models and standards in later phases of SPI.

The LAPPI technique has been validated in over 40 studies conducted in over 30 companies. It is shown to be suitable for modeling the current process and identifying the points of improvement in the process. Motivation monitoring has been applied in one small software company. All case studies are conducted in Finland. Hence, further research is needed to test the suitability of the SPI initiation approach in different cultural environments. In addition, motivation monitoring needs to be applied in additional companies to be able to confirm the benefits it brings.

Bibliography

[1] "The new SME definition: User guide and model declaration," http://ec.europa.eu/enterprise/policies/sme/files/sme_definition/sme_user_guide_en.pdf (2005).

[2] "ISO/IEC 29110 Software engineering – Lifecycle profiles for Very Small Entities (VSEs) ," ISO/IEC 29110 (2011).

[3] A. Valtanen and H.-M. Sihvonen, "Employees' Motivation for SPI: Case Study in a Small Finnish Software Company," in *Software Process Improvement*, Vol. 16, Communications in Computer and Information Science (2008), pp. 152–163.

[4] T. Olson, W. Humphrey, and D. Kitson, "Conducting SEI-assisted software process assessments," Technical Report, CMU/SEI-89-TR-7 (1989).

[5] A. Raninen, J. J. Ahonen, H.-M. Sihvonen, P. Savolainen, and S. Beecham, "LAPPI: A light-weight technique to practical process modeling and improvement target identification," *Journal of Software: Evolution and Process* **25**(9), 915–933 (2013).

[6] W. Humphrey, *Managing the software process* (Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989).

[7] P. Clarke and R. V. O'Connor, "An empirical examination of the extent of software process improvement in software SMEs," *Journal of Software: Evolution and Process* **25**(9), 981–998 (2013).

[8] B. Curtis, M. I. Kellner, and J. Over, "Process Modeling," *Commun. ACM* **35**(9), 75–90 (1992).

[9] S. Zahran, *Software process improvement: practical guidelines for business success* (Addison-Wesley, 1998).

[10] "OECD SME and Entrepreneurship Outlook," Organisation for Economic Co-operation and Development OECD (2005).

[11] M. Rönkkö, O.-P. Mutanen, N. Koivisto, J. Ylitalo, J. Peltonen, A.-M. Touru, S. Hyrynsalmi, P. Poikonen, O. Junna, J. Ali-Yrkkö, A. Valtakoski, Y. Huang, and J. Kantola, *National Software Industry Survey 2008: The Finnish Software Industry in 2007* (Helsinki University of Technology, 2008).

[12] P. Clarke, *Investigating the relationship between software process improvement, situational change, and business success in software SMEs*, PhD thesis (Dublin City University, 2012).

[13] I. Richardson and C. Gresse von Wangenheim, "Guest Editors' Introduction: Why are Small Software Organizations Different?," *IEEE Software* **24**(1), 18 –22 (2007).

[14] P. Kuvaja, "BOOTSTRAP: A software process assessment and improvement methodology," in *Objective Software Quality*, Vol. 926, Lecture Notes in Computer Science (1995), pp. 31–48.

[15] M. C. Paulk, C. V. Weber, B. Curtis, and M. B. Chrissis, *The Capability Maturity Model: Guidelines for Improving the Software Process* (Addison Wesley, Boston, 1995).

[16] "ISO/IEC 15504, Information Technology, Process Assessment, Parts 1-5," ISO/IEC 15504 (2003).

[17] "CMMI for Development, Version 1.3: CMMI-DEV, V1.3, Improving processes for developing better products and services," Technical Report, CMU/SEI-2010-TR-033 (2010).

[18] P. Clarke and R. V. O'Connor, "The influence of SPI on business success in software SMEs: An empirical study," *Journal of Systems and Software* **85**(10), 2356 – 2367 (2012).

[19] G. Canfora, F. García, M. Piattini, F. Ruiz, and C. A. Visaggio, "Applying a framework for the improvement of software process maturity," *Software: Practice and Experience* **36**(3), 283–304 (2006).

[20] A. I. Ferreiro Ferreira, G. Santos, R. Cerqueira, M. Montoni, A. Barreto, A. R. Rocha, A. O. S. Barreto, and R. C. Silva Filho, "ROI of software process improvement at BL informtica: SPI-dex is really worth it," *Software Process: Improvement and Practice* **13**(4), 311–318 (2008).

[21] T. Dybå, "Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context," in *Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering*, ESEC/FSE-11 (2003), pp. 148–157.

[22] I. Richardson, "Software process matrix: a small company SPI model," *Software Process: Improvement and Practice* **6**(3), 157–165 (2001).

[23] M. Staples, M. Niazi, R. Jeffery, A. Abrahams, P. Byatt, and R. Murphy, "An Exploratory Study of Why Organizations Do Not Adopt CMMI," *Journal of Systems and Software* **80**(6), 883 – 895 (2007).

[24] R. V. Horvat, I. Rozman, and J. Györkös, "Managing the complexity of SPI in small companies," *Software Process: Improvement and Practice* **5**(1), 45–54 (2000).

[25] F. McCaffery, P. Taylor, and G. Coleman, "Adept: A unified assessment method for small software companies," *IEEE Software* **24**(1), 24–31 (2007).

[26] C. von Wangenheim, A. Anacleto, and C. Salviano, "Helping small companies assess software processes," *IEEE Software* **23**(1), 91 – 98 (2006).

[27] S. Trudel, J.-M. Lavoie, M.-C. Par, and W. Suryn, "PEM: The small company-dedicated software process quality evaluation method combining CMMISM and ISO/IEC 14598," *Software Quality Journal* **14**(1), 7–23 (2006).

[28] A. Cater-Steel, M. Toleman, and T. Rout, "Process improvement for small firms: An evaluation of the RAPID assessment-based method," *Information and Software Technology* **48**(5), 323 – 334 (2006).

[29] H. Conradi and A. Fuggetta, "Improving software process improvement," *IEEE Software* **19**(4), 92 – 99 (2002).

[30] F. McCaffery and G. Coleman, "Lightweight SPI assessments: what is the real cost?," *Software Process: Improvement and Practice* **14**(5), 271–278 (2009).

[31] A. Raninen, H. M. J. J. Ahonen, and S. Beecham, "Applying Software Process Modeling to Improve Customer Support Processes," *Submitted to: Journal of Software: Evolution and Process* **?**(?), ? (2013).

[32] B. McFeeley, "IDEAL: A User′s Guide for Software Process Improvement.," Technical report, http://www.sei.cmu.edu/reports/96hb001.pdf (1996).

[33] T. Toroi, A. Raninen, and L. Väätäinen, "Identifying Process Improvement Targets in Test Processes: A Case Study," in *Software Maintenance (ICSM), 2013 29th IEEE International Conference on* (2013), pp. 11–19.

[34] N. Baddoo and T. Hall, "Motivators of Software Process Improvement: an analysis of practitioners' views," *Journal of Systems and Software* **62**(2), 85 – 96 (2002).

[35] A. Rainer and T. Hall, "Key success factors for implementing software process improvement: a maturity-based analysis," *Journal of Systems and Software* **62**(2), 71 – 84 (2002).

[36] C. Laporte, S. Alexandre, and A. Renault, "Developing International Standards for Very Small Enterprises," *Computer* **41**(3), 98 –101 (2008).

[37] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and Software Technology* **50**(910), 833 – 859 (2008).

[38] J. Erickson, K. Lyytinen, and K. Siau, "Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research," *Journal of Database Management* **16**(4), 88–100 (2005).

[39] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of Migrating to Agile Methodologies," *Commun. ACM* **48**(5), 72–78 (2005).

[40] T. Dybå, "Improvisation in Small Software Organizations," *Software, IEEE* **17**(5), 82–87 (2000).

[41] C. Jones, "Patterns of large software systems: failure and success," *Computer* **28**(3), 86 – 87 (1995).

[42] K. Moløkken and M. Jørgensen, "A review of software surveys on software effort estimation," in *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on* (2003), pp. 223 – 230.

[43] C. Halvorsen and R. Conradi, "A Taxonomy to Compare SPI Frameworks," in *Software Process Technology*, Vol. 2077, V. Ambriola, ed. (Springer Berlin / Heidelberg, 2001), pp. 217-235.

[44] J. Herbsleb, A. Carleton, J. Rozum, J. Siegel, and D. Zubrow, "Benefits of CMM-based software process improvement: Initial results," Technical report, CMU/SEI-94-TR-13 (1994).

[45] R. Achatz and F. Paulisch, "Industrial strength software and quality: software and engineering at Siemens," in *Quality Software, 2003. Proceedings. Third International Conference on* (2003), pp. 321 – 326.

[46] D. Goldenson and D. Gibson, "Demonstrating the impact and benefits of CMMI: an update and preliminary results," Technical report, CMU/SEI-2003-SR-009 (2003).

[47] F. Pino, C. Pardo, F. García, and M. Piattini, "Assessment methodology for software process improvement in small organizations," *Information and Software Technology* **52**(10), 1044–1061 (2010).

[48] "ISO/IEC 33012: Information technology - Process assessment - Guide for process improvement," ISO/IEC TC JTC1/SC 7/WG 10 (2009).

[49] "IEEE Standard Glossary of Software Engineering Terminology," IEEE Std 610.12-1990, The Institute of Electrical and Electronics Engineers (1990).

[50] "ISO/IEC Guide 2:2004 Standardization and related activities - General vocabulary," ISO/IEC (2004).

[51] http://dictionary.reference.com/browse/de+facto+standard (valid 5.9.2010).

[52] http://oxforddictionaries.com/ (valid 14.3.2011).

[53] S. Sheard, "Evolution of the frameworks quagmire," *Computer* **34**(7), 96–98 (2002).

[54] J. Juran, A. Godfrey, and D. Williams, *Juran's quality handbook*, Vol. 2, (McGraw-Hill London, 1999).

[55] "ISO 9000 Quality management systems – Fundamentals and vocabulary.," ISO 9000 (2005).

[56] M. Harry and R. Schroeder, *Six sigma: the breakthrough management strategy revolutionizing the world's top corporations* (Broadway Business, 2000).

[57] B. Mutafelija and H. Stromberg, *Systematic Process Improvement Using Iso 9001: 2000 And Cmmism* (Artech House, 2003).

[58] M. Lepmets, E. Ras, and A. Renault, "Organizational Support for Process Improvement Results of an International Survey," in *Software Process Improvement and Capability Determination*,

Vol. 155, R. V. OConnor, T. Rout, F. McCaffery, and A. Dorling, eds. (Springer Berlin Heidelberg, 2011), pp. 133-144.

[59] “ISO/IEC 12207, 2008. Software Life Cycle Processes.,” ISO/IEC 12207 (2008).

[60] “ISO/IEC 15289, 2006. Systems and software engineering – Content of systems and software life cycle process information products.,” ISO/IEC 15289 (2006).

[61] “ISO/IEC 15288 Systems and software engineering – System life cycle processes ,” ISO/IEC 15288 (2008).

[62] M. Chrissis, M. Konrad, and S. Shrum, *CMMI Second Edition: Guidelines for Process Integration and Product Improvement (Version 1.2)* (Addison-Wesley, 2007).

[63] “CMMI Appraisals,” http://cmmiinstitute.com/cmmi-solutions/cmmi-appraisals/ (2013).

[64] “Transition from ISO/IEC 15504 to ISO/IEC 330xx,” ISO/IEC JTC1 SC7 WG10 (2013).

[65] “ISO/IEC TR 33014:2013 Information technology – Process assessment – Guide for process improvement,” ISO/IEC 33014 (2013).

[66] “ISO/IEC 15504 Information technology - Process assessment - Part 2: Performing an Assessment ,” ISO/IEC 15504 (2003).

[67] T. Kaltio and A. Kinnula, “Deploying the defined SW process,” *Software Process: Improvement and Practice* **5**(1), 65–83 (2000).

[68] N. Baddoo and T. Hall, “De-motivators for software process improvement: an analysis of practitioners’ views,” *Journal of Systems and Software* **66**(1), 23 – 33 (2003).

[69] M. Niazi and M. Babar, “De-motivators of Software Process Improvement: An Analysis of Vietnamese Practitioners

Views," in *Product-Focused Software Process Improvement*, Vol. 4589, Lecture Notes in Computer Science (2007), pp. 118–131.

[70] M. Niazi and M. Ali Babar, "Motivators of Software Process Improvement: An Analysis of Vietnamese Practitioners' Views," in *Proceedings of 11th International Conference on Evaluation and Assessment in Software Engineering (EASE)* (2007).

[71] T. Hall, D. Jagielska, and N. Baddoo, "Motivating developer performance to improve project outcomes in a high maturity organization," *Software Quality Journal* **15**(4), 365–381 (2007).

[72] S. Garcia, "Thought on Applying CMMI in Small Settings," http://www.sei.cmu.edu/library/assets/garcia-thoughts1.pdf (2005).

[73] P. Allen, M. Ramachandran, and H. Abushama, "PRISMS: an approach to software process improvement for small to medium enterprises," pp. 211 – 214 (2003).

[74] F. Pino, F. García, and M. Piattini, "Software process improvement in small and medium software enterprises: a systematic review," *Software Quality Journal* **16**, 237–261 (2008).

[75] K. Kautz, "Software process improvement in very small enterprises: does it pay off?," *Software Process: Improvement and Practice* **4**(4), 209–226 (1998).

[76] M. Niazi, D. Wilson, and D. Zowghi, "Critical success factors for software process improvement implementation: an empirical study," *Software Process: Improvement and Practice* **11**(2), 193–211 (2006).

[77] D. Stelzer and W. Mellis, "Success factors of organizational change in software process improvement," *Software Process: Improvement and Practice* **4**(4), 227–250 (1998).

[78] T. Rout, A. Tuffley, B. Cahill, and B. Hodgen, "The RAPID assessment of software process capability," in *Proceedings of 1st International SPICE Conference* (Citeseer, 2000), pp. 47–56.

[79] C. Wangenheim, S. Weber, J. Hauck, and G. Trentin, "Experiences on establishing software processes in small companies," *Information and Software Technology* **48**(9), 890–900 (2006).

[80] T. Dingsøyr, N. Moe, T. Dybå, and R. Conradi, "A Workshop-Oriented Approach for Defining Electronic Process Guides," in *Software Process Modeling*, Vol. 10, International Series in Software Engineering (2005), pp. 187–205.

[81] M. Poppendieck, "Lean Software Development," in *ICSE COMPANION '07: Companion to the proceedings of the 29th International Conference on Software Engineering* (2007), pp. 165–166.

[82] K. Petersen and C. Wohlin, "Software process improvement through the Lean Measurement (SPI-LEAM) method," *Journal of Systems and Software* **83**(7), 1275 – 1287 (2010), SPLC 2008.

[83] I. Richardson, "SPI Models: What Characteristics Are Required for Small Software Development Companies?," in *Software Quality - ECSQ 2002*, Vol. 2349, Lecture Notes in Computer Science (2006), pp. 100–113.

[84] A. Anacleto, C. G. V. Wangenheim, C. Clênio F. Salviano, and R. Savi, "A Method for Process Assessment in Small Software Companies," in *4th International SPICE Conference on Process Assessment and Improvement* (2004).

[85] S. Basri and R. V. OConnor, "A study of software development team dynamics in SPI," in *Systems, Software and Service Process Improvement* (2011), pp. 143–154.

[86] S. Ryan and R. V. OConnor, "Development of a team measure for tacit knowledge in software development teams," *Journal of Systems and Software* **82**(2), 229–240 (2009).

[87] J. J. Ahonen, M. Forsell, and S. Taskinen, "A modest but practical software process modeling technique for software pro-

cess improvement," *Software Process: Improvement and Practice* **7**(1), 33 – 44 (2002).

[88] A. Valtanen and J. Ahonen, "Big Improvements with Small Changes: Improving the Processes of a Small Software Company," in *Product-Focused Software Process Improvement*, Vol. 5089, Lecture Notes in Computer Science (2008), pp. 258–272.

[89] G. M. Giaglis, "A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques," *International Journal of Flexible Manufacturing Systems* **13**(2), 209–228 (2001).

[90] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Languange* (Addison-Wesley Professional, 2004).

[91] M. I. Kellner, R. J. Madachy, and D. M. Raffo, "Software process simulation modeling: Why? What? How?," *Journal of Systems and Software* **46**(23), 91 – 105 (1999).

[92] J. L. Jones, *Structured programming logic: a flowcharting approach* (Prentice-Hall, Inc., 1985).

[93] T. Huckvale and M. Ould, *Process Modeling – Who, What and How: Role Activity Diagramming* (Idea Group Publishing, Harrisburg, PA, 1995).

[94] J. Ahonen, H. Lintinen, and S.-K. Taskinen, "Improving the Reuse Process is Based on Understanding the Business and the Products: Four Case Studies," in *Product Focused Software Process Improvement*, Vol. 2559, M. Oivo and S. Komi-Sirvi, eds. (Springer Berlin Heidelberg, 2002), pp. 273-290.

[95] R. L. Baskerville, "Investigating information systems with action research," *Communications of the AIS* **2**(3es), 2 – 32 (1999).

[96] R. Yin, *Case study research, Design and methods* (Sage, 2009).

[97] A. Oyegoke, "The constructive research approach in project management research," *International Journal of Managing Projects in Business* **4**(4), 573–595 (2011).

[98] D. Greenwood and M. Levin, *Introduction to action research: Social research for social change* (Sage Publications, Inc, 2007).

[99] I. Benbasat, D. K. Goldstein, and M. Mead, "The Case Research Strategy in Studies of Information Systems," *MIS Quarterly* **11**(3), pp. 369–386 (1987).

[100] D. E. Avison, F. Lau, M. D. Myers, and P. A. Nielsen, "Action research," *Communications of the ACM* **42**(1), 94–97 (1999).

[101] D. Sjøberg, T. Dybå, and M. Jørgensen, "The Future of Empirical Methods in Software Engineering Research," in *Future of Software Engineering, 2007. FOSE '07* (2007), pp. 358 –378.

[102] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering* **14**(2), 131–164 (2009).

[103] B. Kitchenham, L. Pickard, and S. L. Pfleeger, "Case studies for method and tool evaluation," *IEEE Software* **12**(4), 52–62 (1995).

[104] C. Wohlin, M. Höst, and K. Henningsson, "Empirical Research Methods in Software Engineering," in *Empirical Methods and Studies in Software Engineering*, Vol. 2765, R. Conradi and A. Wang, eds. (Springer Berlin Heidelberg, 2003), pp. 7-23.

[105] S. Kvale, *Interviews: An introduction to qualitative research interviewing* (Sage Publications, Inc, 1996).

[106] L. Rea, R. Parker, and R. Allen, *Designing and conducting survey research: A comprehensive guide* (Jossey-Bass Publishers San Francisco, CA, 1992).

[107] S. A. Kasanen E, Lukka K, "The constructive approach in management accounting research," *Journal of Management Accounting Research* (5), 243–264 (1993).

[108] K. Vlaanderen, I. van de Weerd, and S. Brinkkemper, "Improving software product management: a knowledge management approach," *International Journal of Business Information Systems* **12**(1), 3–22 (2013).

[109] B. McFeeley, "IDEAL: A User's Guide for Software Process Improvement.," (1996).

[110] D. R. Denison and G. M. Spreitzer, "Organizational culture and organizational development: A competing values approach," *Research in organizational change and development* **5**(1), 1–21 (1991).

[111] M. Unterkalmsteiner, T. Gorschek, A. Islam, C. K. Cheng, R. Permadi, and R. Feldt, "Evaluation and Measurement of Software Process Improvement: A Systematic Literature Review," *Software Engineering, IEEE Transactions on* **38**(2), 398–424 (2012).

Anu Raninen

Practical Process Improvement

How to Initiate Software Process Improvement in a Small Company

This thesis provides a validated, practical, and easy to apply approach for small software companies to cost-effectively initiate Software Process Improvement (SPI). The research is driven by close collaboration with Small or Medium-sized Enterprises (SMEs) undergoing SPI. The suggested approach consists of the LAPPI technique, a light-weight technique to practical process modeling and improvement target identification. LAPPI is complemented by research into motivation monitoring, a recommended practice that supports the initiation and success of SPI initiatives.

UNIVERSITY OF EASTERN FINLAND

Publications of the University of Eastern Finland

Dissertations in Forestry and Natural Sciences

ISBN 978-952-61-1429-3